WEAKLY SUPERVISED LEARNING FOR MUSICAL INSTRUMENT CLASSIFICATION

A Dissertation
Presented to
The Academic Faculty

By

Siddharth Kumar Gururani

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Music

Georgia Institute of Technology

December 2020

WEAKLY SUPERVISED LEARNING FOR MUSICAL INSTRUMENT CLASSIFICATION

Approved by:

Dr. Alexander Lerch
School of Music
Georgia Institute of Technology

Dr. Jason Freeman
School of Music
Georgia Institute of Technology

Dr. Devi Parikh
School of Interactive Computing
Georgia Institute of Technology

Markus Cremer
Vice President, Applied Research
Gracenote, Inc.

Dr. Gil Weinberg
School of Music
Georgia Institute of Technology

Date Approved: August 3, 2020

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my thesis advisor, Prof. Alexander Lerch, for his constant support, patience, and valuable guidance throughout my time at Georgia Tech. Prof. Lerch gave me the opportunity to join his lab five years ago and has since been a mentor and a friend, helping me develop into a much-improved version of myself. I am eternally grateful to him for instilling in me an academic curiosity and the zeal to keep on learning. I look forward to future collaborations with him and his students.

I also thank the members of my thesis committee. Particularly, I would like to thank Prof. Devi Parikh for her valuable insights during the formulation of the thesis proposal, and for her feedback on my final thesis. I thank Prof. Jason Freeman and Prof. Gil Weinberg for their unique perspectives that helped me consider the broader impact of my research. Both Prof. Freeman and Prof. Weinberg have also helped me develop a well-rounded understanding of music technology through the various courses they taught me at GTCMT. I am also grateful to Markus Cremer and the Applied Research group at Gracenote Inc.: Bob Coover, and Joe Renner for the regular discussions and feedback which have been valuable for shaping the research in this thesis.

Next, I would like to thank my collaborators: Cameron Summers, and Mohit Sharma for their help with my research. The papers co-authored with them are important parts of this thesis.

This acknowledgement section is incomplete without mentioning the people who played an important role in guiding me to pursue a Ph.D. in Music Technology. I thank my Masters' project advisor, Prof. Sourangshu Bhattacharya, and my advisor while I was a visiting scholar at HKUST, Prof. Pascale Fung, who introduced me to the field of music information retrieval and gave me the opportunity to learn from them.

I am also thankful to several people who contributed immensely towards making the typically stressful graduate school experience incredibly enjoyable. The friends I have made

at Georgia Tech, summer internships, and several ISMIRs are all responsible, in one way or another, for helping me through these five years. I would like to especially mention Ashis, Amruta, Vinod, Chih-Wei, Mohit, and Akanksha who spent a significant amount of time during the final months of my Ph.D. in helping me with the thesis and subsequent defense.

Last but most importantly, I would like to thank my parents and my brother for supporting me throughout my journey, especially at times where I might have been unbearable. This thesis would not exist without their unconditional love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Automatically recognizing musical instruments in audio recordings is an important task in music information retrieval (MIR). With increasing complexity of modeling techniques, the focus of the Musical Instrument Classification (MIC) task has shifted from single note audio analysis to MIC with real world polytimbral music. Increasingly complex models also increase the need for high quality labeled data. For the MIC task, there do not exist such large-scale fully annotated datasets. Instead researchers tend to utilize multi-track data to obtain fine-grained instrument activity annotation. Such datasets are also known as strongly labeled datasets (SLDs). These datasets are usually small and skewed in terms of genre and instrument distribution. Hence, SLDs are not the ideal choice for training generalizable MIC models.

Recently, weakly labeled datasets (WLDs), with only clip-level annotations, have been presented. These are typically larger in scale than strongly labeled datasets (SLDs). However, methods popular in MIC literature are designed to be trained and evaluated SLDs. These do not naturally extend to the task of weakly labeled MIC. Additionally, during the labeling process, clips are not necessarily annotated with a class label for each instrument. This leads to missing labels in the dataset making it a partially labeled dataset.

In this thesis, three methods are proposed to address challenges posed by weakly labeled and partially labeled data. The first one aims at learning using weak labels. The MIC task is formulated as a multi-instance multi-label classification problem. Under this framework, an attention-based model is proposed that can focus on salient instances in weakly labeled data. The other two methods focus on utilizing any information that may be gained from data with missing labels. These methods fall under the semi-supervised learning (SSL) framework, where models are trained using labeled and unlabeled data. The first semi-supervised method involves deep generative models that extend the unsupervised variational autoencoder to a semi-supervised model. The final method is based on consistency regularization-based SSL. The method proposed uses the mean teacher model, where a teacher model maintains a moving average or low-pass filtered version of a student model. The consistency

regularization loss is unsupervised and may thus be applied to both labeled and unlabeled data. Additional experiments on music tagging with a large-scale WLD demonstrates the effectiveness of consistency regularization with limited labeled data.

The methods presented in this thesis generally outperform methods developed using SLDs. The findings in this thesis not only impact the MIC task but also impact other music classification tasks where labeled data might be scarce. This thesis hopes to pave the way for future researchers to venture away from purely supervised learning and also consider weakly supervised approaches to solve MIR problems without access to large amounts of data.

CHAPTER 1
INTRODUCTION

Music is a work of art where the primary medium of communication is sound. In the most common form of musical communication, the composer produces a composition which is rendered into a sonic realization by a performer to be then perceived and interpreted by the listener [1]. The physical characteristics of sound: frequency, amplitude, duration, and form are sensed by the listener as perceived characteristics: pitch, loudness, time, and timbre [2]. The human auditory system enables us to understand musical concepts such as rhythm, beats, and harmony, without so much as an active thought. We can follow along and reproduce different rhythms [3], detect phrase boundaries [4], separate tones of different instrument [5], and so on. These tasks can be managed by most people and often do not require musical training.

The identification of musical instruments relies on the human ability to distinguish between different sources of sound. *Timbre* is the property of sound that plays an important role in the recognition of different sound sources. The Acoustical Society of America (ASA) defines timbre as: "that attribute of auditory sensation which enables a listener to judge that two nonidentical sounds, similarly presented and having the same loudness and pitch, are dissimilar" [6]. This subtractive definition has often been criticized due to the absence of any perceptual meaning being attributed to timbre. Research involving resynthesis of instrument sounds with changes or simplification to the spectrum established that the spectrum of sound produced by objects plays a major role in the perception of timbre [7, 8]. Listeners were able to correctly discriminate instrument sounds for attacks such as spectral envelope smoothing. However, frequency-variation smoothing, and frequency flattening led to poorer discriminative ability.

In the quest to fully understand the dimensions of the timbre space, it was also found that

the temporal envelope of sound played a role in timbre perception [9]. The temporal envelope of sound can be characterized by the attack (onset), decay (drop in energy immediately following the onset), sustain (steady state) and release (offset). Experiments involving modifications to the envelope confirmed the importance of these characteristics, especially the attack and sustain [10, 11, 12] thus leading to the conclusion that timbre has both spectral and temporal dimensions. Musical instruments occupy different locations in this perceptual timbre space, with the distances between them in this space largely dependent on their physical properties [13]. These distances in the timbre space correlate to humans' abilities to distinguish and recognize different musical instruments.

With the rise of computational methods, researchers attempted to build computer systems capable of audio signal analysis by combining knowledge gained from psychoacoustics, Digital Signal Processing (DSP) and Machine Learning (ML). The field of computational auditory scene analysis (CASA) generally aims to develop methods capable of understanding input streams of audio, with widespread impact on speech processing and music analysis. Automatic identification of musical instruments was one of the early problems in CASA with published research dating back to 1995 [14]. Motivated by the need to explain and computationally model human perception of musical sounds, several studies utilized auditory-based features for instrument recognition, achieving encouraging results for single note instrument sounds [15, 16, 17].

Fast-forward to the present day, when music streaming services such as Spotify, Apple Music, and Pandora have made music significantly more accessible. Computational music analysis is a dedicated field of research known as Music Information Retrieval (MIR). The MIR community brings together cutting-edge DSP and ML research to analyze, understand, and even generate music. Systems for MIC, a task widely regarded as an important MIR problem, saw improvements largely due to increased availability of data and improved methods for audio analysis. However, as researchers aimed to build systems for MIC involving more complex audio signals, several issues pertaining to music data came to the

forefront. In this thesis, I expand upon these data challenges and present methods to tackle them in the context of music instrument classification. The methods described in this thesis are not specific to instrument classification and can easily be generalized to other MIR and audio analysis tasks.

1.1 Motivation

Technological advancement has led to music creation, distribution, and consumption at a massive scale. The ubiquitous nature of music drives the need to push the boundaries of music technology even further. My research is motivated by the desire to improve audio content analysis systems thus enabling machines to extract information from complex musical audio signals. In particular, I am interested in creating systems capable of identifying the musical instruments in a given music audio signal. While methods for instrument classification of single-note recordings are fairly successful, they are unable to perform well for recordings of music which contain multiple instruments (see Section 2.1). Compared to single-note and single-instrument music, music with multiple instruments is more challenging to analyze and has been the focus of instrument classification research in the MIR community recently [18, 19, 20, 21, 22]. The superposition of multiple instruments in time and frequency and the variance of timbre and playing techniques for the same instrument are a few of the challenges with multi-instrument music.

This thesis is driven by the need to address certain data challenges in MIC and the MIR community as a whole. Deep learning methods continue to grow more powerful and popular in domains like computer vision, speech processing and natural language processing. As the MIR community is quick paced and adopts new modeling methods promptly, there is a growing need for large-scale training data. Instead, the community faces a dearth of large-scale datasets with only a handful of labeled datasets at a scale comparable to datasets in vision or speech recognition. Collecting large-scale annotated data often requires trained experts, is cost intensive, and is outpaced by the increasing need for data. Therefore, the

most viable solution is to develop methods capable of leveraging few labeled data and larger amounts of unlabeled data.

1.2 Applications

An additional factor motivating this thesis is the wide impact the MIC task has on the broader field of music technology. Systems for MIC are applicable in several downstream MIR tasks, as well as in direct consumer applications. The following non-exhaustive list of applications shows how influential research in MIC has the potential to be.

1.2.1 Music Recommendation

Given the scale of music libraries accessible directly on smartphones via streaming services, the quality of music recommendation makes or breaks the music listening experience for the end-user. Collaborative filtering-based recommender systems [23, 24] are most commonly used in present day streaming services, which utilize user listening or preference data. One of the major drawbacks of such systems is the lack of user preference data for new songs. This problem is popularly known as the cold-start problem [25].

One method to address the cold-start problem is to use content-based recommendation systems along with collaborative filtering. Such systems rely on information extracted from, or metadata associated with new tracks to compute similarity scores to existing tracks in the library which enables the system to recommend these new tracks [26, 27, 28]. This metadata may contain information such as the genre, artist, tempo, instruments, mood, etc. The best example of metadata-based recommendation in practice is Pandora — a popular music streaming service. The Music Genome Project [1] at Pandora utilized experts annotate large catalog of music with 450 music 'genes' or features. Similarity between tracks can then be computed using this 450-dimensional feature vector [29].

One potential challenge with expert annotations is the inhibitive cost. Thus, methods to

[1] https://www.pandora.com/about/mgp (Last accessed: 7/12/2020)

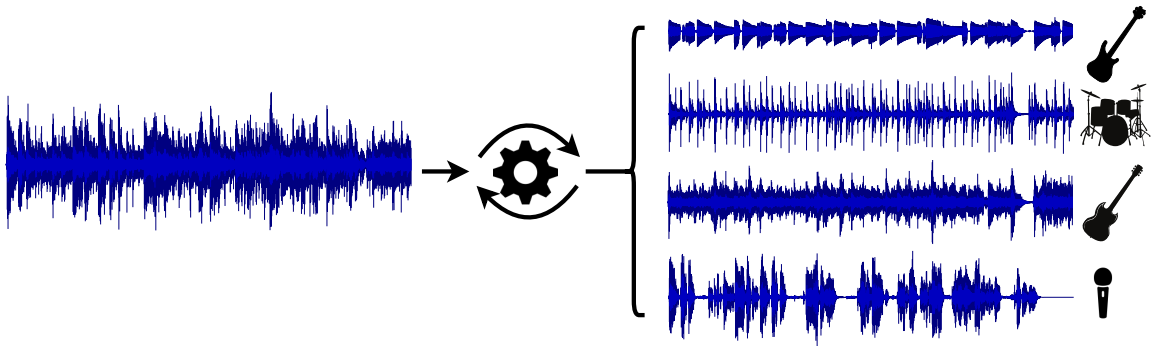

Figure 1.1: Music source separation: Methods for source separation process a mixture of sources to extract individual source audio as shown. The input song on the left contains bass, drums, guitars, and vocals, which are extracted by the source separation system. These systems are evaluated using metrics such as source-to-distortion ratio (SDR), source-to-interference ratio (SIR), and source-to-artifact ratio (SAR). SDR measures the amount of distortion the system introduces to the individual source audio. SIR measures the interference from other other sources. SAR measures the number of sonic artifacts introduced by the system.

automatically extract relevant metadata are increasingly being researched [30, 31]. Instrumentation metadata is especially important since instruments have a strong influence on the overall timbre of a song. Additionally, availability of instrumentation allows recommender systems to learn user preference in terms of instruments and consequently recommend music based on instrument preference.

1.2.2 Music Source Separation

Music source separation is the task of extracting audio signals of specific sources from a mixture of sources. Figure 1.1 depicts the source separation task. While a lot of research has been conducted in blind music source separation [32, 33, 34], there are also several works utilizing additional information such as score-informed source separation [35, 36, 37, 38].

Recent work studied the impact of instrument or vocal activity on source separation performance [39, 40, 41, 42, 43]. Hung and Lerch used multi-task learning to train a source separation model jointly with an instrument activity detection model and demonstrate improved performance for source separation [39]. Slizovskaia et al. utilized instrument

activity conditioning for source separation and found the method to be effective for cases where the number of sources is greater than 2 [43]. Similarly, Swaminathan and Lerch used vocal activity as an additional input to a vocal separation model and found the separation performance to improve across all metrics [40]. They also utilized a pre-trained vocal activity detection network to extract vocal activity which is subsequently used as input to the separation model. Stoller et al. utilized a multi-task learning approach to jointly predict vocal activity and separate vocals [41]. They found improvement in performance for both tasks. These studies demonstrate the usefulness of MIC in the source separation task.

1.2.3 Music Transcription

Note that the metadata extracted for recommendation systems typically represent higher-level information about the corresponding music, such as genre or mood. However, one of the main goals of MIR is the development automatic music transcription (AMT) systems [44], which extract lower-level information from musical audio, such as pitch, note onsets, and their duration. One example of an ideal AMT system takes as input a musical audio signal and outputs a symbolic representation of the performed music such as the musical score or tablature. Figure 1.2 illustrates AMT systems from a high-level. AMT is particularly useful in music creation, music education, archival activities for old records, as well as studies for improvisatory music performance.

The AMT problem can be broken down into the following sub-tasks: [45]

(i) Pitch tracking

(ii) Key detection

(iii) Chord detection

(iv) Beat and downbeat tracking

(v) Rhythm tracking

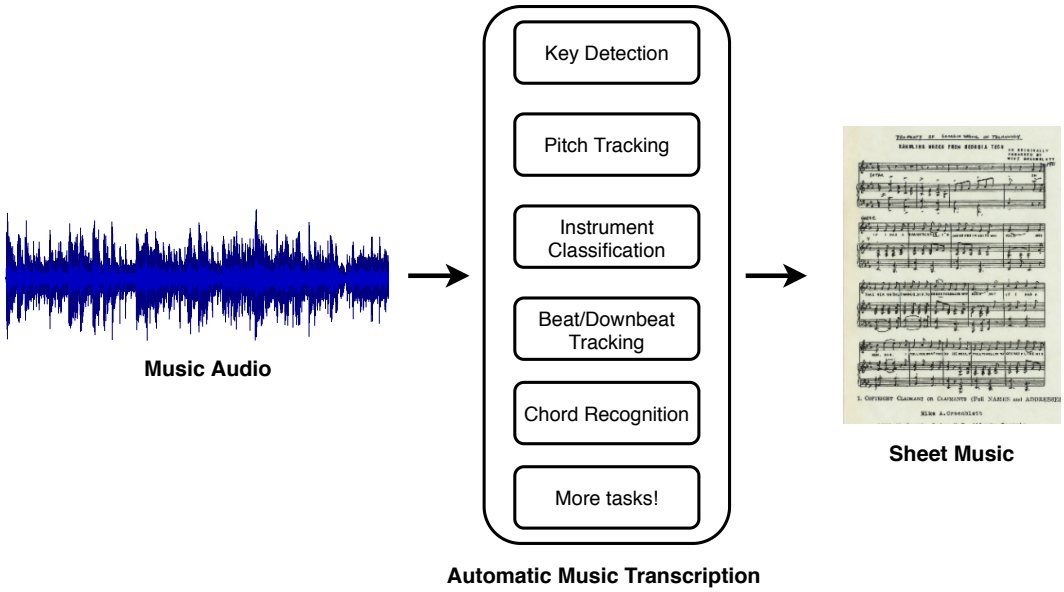

Figure 1.2: Automatic Music Transcription: The goal is to go from audio to a symbolic representation of the music in the audio, in this case, sheet music. The task involves several sub-tasks. These tasks include, but are not limited to, pitch tracking, tempo estimation, beat or downbeat detection, key detection, chord detection, instrument classification, and structure analysis. Each of these sub-tasks are important in obtaining information that is vital to the final transcription.

(vi) Onset detection

(vii) Instrument or voicing detection

(viii) Structure analysis

(ix) Expressive timing and ornamentation detection

The methods addressing the aforementioned sub-problems may be combined in various ways to achieve an AMT system. Current research on AMT typically involves single instrument recordings, with most of the work focusing on piano [46, 47]. This, however, does not mean that AMT literature is only limited to piano recordings. While there exist recent works that focus on music with multiple instruments, they are rare and restricted to classical music recordings [48].

1.2.4 Automatic Drum Transcription

Automatic drum transcription (ADT) is an MIR task that shares commonalities with both AMT as well as MIC. The task deals with transcribing drums in both isolated and multi-instrument recordings [49]. While the end product of ADT is a transcription of the drum performance, the ADT task can be considered a special case of instrument classification. This is so because the drum-kit consists of several different instruments and the transcription task can essentially be converted into an activity detection problem. Thus methods for MIC may very well be applied to the task of ADT.

1.2.5 Music Search and Retrieval

The usefulness of semantic metadata is not restricted to music recommendation systems or being able to use instrument search terms for music. The vast libraries of music that are now very easily accessible through mobile devices need to be have intelligent search capabilities. Systems capable of extracting instrument activity in songs would allow for interesting opportunities in music search. For example, users who may be interested in listening to a guitar or saxophone solo gain the ability to search *within* songs for the regions of interest. Additionally, user-interfaces for music listening can be adapted based on instrument activity, as illustrated by [50]. Instrument-based music search would also prove useful for music creators. Organizing and searching music samples in large sample libraries can benefit from MIC research by allowing creators to sort by instruments, for example.

1.2.6 Multi-Task Learning in MIR

Multi-task learning (MTL) is a ML concept in which models are supervised using data for multiple related tasks. The idea is that a shared model for two related tasks generalizes better due to the presence of different but related supervisory signals from the same domain [51]. The reason this works is the shared representation that is learned for multiple tasks incorporates domain-specific inductive bias that is helpful for all tasks the model is

being trained for.

MTL has been widely adopted in MIR literature for several tasks besides source separation (see Section 1.2.2), such as: tempo, beat, and downbeat estimation, transcription, and drum transcription [52, 46, 53, 54]. Similarly, MIC or instrument activity detection has be used in an MTL setting combined with pitch detection [55]. MIC can be combined with other related MIR tasks as well, such as genre classification, mood classification, automatic tagging, and music transcription. Generally speaking, the field of MIR consists of several inter-related MIR tasks. The MTL framework may be applied to develop networks of MIR tasks leading to large monolithic models that solve multiple tasks with the potential to outperform individual models for each task.

1.3 Data Challenges

The research workflow for MIC follows a fairly standard template. Initially, a dataset of music is collected, and a set of instruments is chosen. The music is then annotated with instruments present or absent. The next phase typically involves feature extraction: conversion of raw audio data to a representation suitable for ML models to ingest. This may involve spectro-temporal audio features [56, 57] or a deep neural network-based acoustic model which learns features from the raw data [18, 20]. The dataset is then divided into a training and testing set. The processed training data is used to train a model to infer the instruments present or absent. Finally, performance of the model is evaluated using the testing set.

This workflow is essentially identical for most supervised learning problems across different application domains like automatic speech recognition, computer vision, and natural language processing. With the advancement in deep learning in recent years, most applied research involves data preprocessing, followed by constructing a deep neural network incorporating desired inductive biases which is then trained to perform the task at hand. As mentioned briefly in Section 1.1, one of the key factors in the effectiveness of deep learning

has been the scale at which data is now available. Investment in large-scale data collection and data annotation has benefited domains like computer vision and speech recognition, where DNNs are now deployed in production-level services[2].

MIR as a field, on the other hand, has yet to grow to a stage where there are several large-scale audio datasets available for various tasks within the field. With the exception of automatic tagging [58] and piano transcription [59], most tasks in MIR are accompanied by smaller scale publicly available datasets. The MIC task faces the same challenge of having limited labeled data. The obvious approach to overcome this challenge is to collect more large-scale annotated data. However, collecting large-scale annotated data is not as straightforward as simply crowd-sourcing new annotations. Annotators are required to be have some degree of music proficiency. The need for music experts and the fact that manual audio tagging is involved makes the process expensive, both in terms of time and money, thus making this approach difficult.

Another approach to overcome the aforementioned challenge is to investigate methods that can learn effectively with limited labeled data. MIC research has almost exclusively utilized supervised learning methods (see Section 2.1). However, ML algorithms are not limited to supervised learning with labeled data. ML researchers have spent decades developing methods to learn using data with missing labels and unlabeled data, which led to semi-supervised and unsupervised learning methods [60, 61, 62, 63]. Despite labeled music datasets being small in scale, the abundance of unlabeled music data that is available may indeed prove useful. The methods proposed in this thesis are an attempt to shed some light on the applicability of methods that do not solely rely on labeled data, in the hopes of steering the focus of research in MIR away from strongly supervised towards weakly supervised methods.

[2]https://azure.microsoft.com/en-us/services/cognitive-services/ (Last Accessed: 7/13/2020)

1.4 Outline

The first two chapters of this thesis set the background for musical instrument classification, introduce the challenges that this thesis aims to address, and frame the research questions. Chapter 2 also presents and compares models for strongly supervised instrument classification discussing various issues with the experiment setup and data which eventually led to the experiments with weakly labeled data. Chapter 3 addresses one of the data challenges: learning from weakly labeled audio data. This chapter pertains to the first research question, and focuses on an attention-based model that uses adaptive aggregation of model predictions to infer the final labels for input audio. chapters 4 and 5 address the second challenge: learning from missing labels or unlabeled data. These chapters answer the second and third research questions respectively. These research questions pertain to the usage of generative models in MIC and the impact of unlabeled data. Both of the chapters present methods for semi-supervised learning, with generative models for semi-supervised learning being the focus of chapter five, and consistency regularization-based semi-supervised learning being the focus of chapter six. The seventh and final chapter brings the conclusions from the three broad experiments together and discusses future directions of research. Figure 1.3 depicts the structure of this thesis diagrammatically.

The contents of this thesis have been published in the following publications:

- Siddharth Gururani and Alexander Lerch, "Mixing Secrets: A Multi-Track Dataset for Instrument Recognition in Polyphonic Music," in Late Breaking Demo (Extended Abstract), Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), 2017.

- Siddharth Gururani, Cameron Summers, and Alexander Lerch, "Instrument Activity Detection in Polyphonic Music using Deep Neural Networks" in Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), 2018.

- Siddharth Gururani, Mohit Sharma, and Alexander Lerch, "An Attention Mechanism

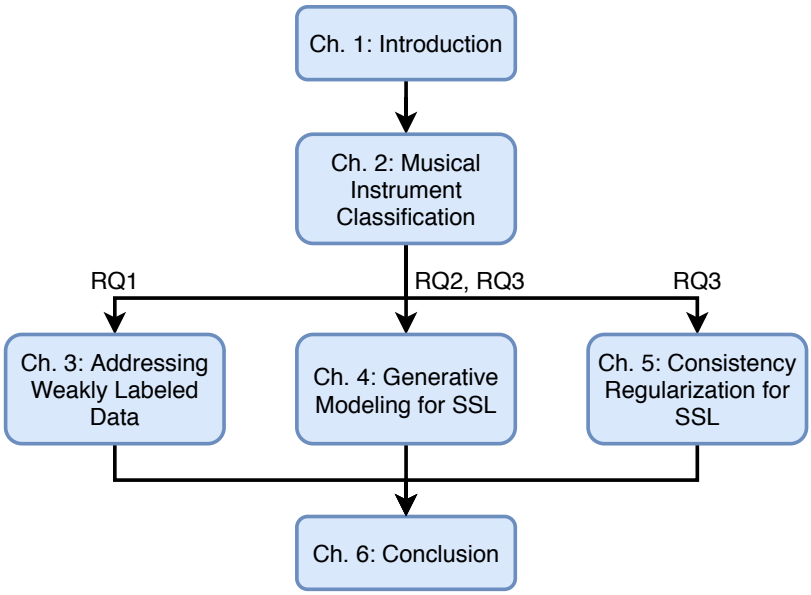

Figure 1.3: High-level outline of the thesis

for Music Instrument Recognition" in Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), 2019.

Chapter 2 introduces the task of MIC in detail and provides a background of the variants of MIC followed by a thorough literature survey. Section 2.2 describes various datasets that are used for MIC, followed by a discussion on the evaluation metrics used to measure model performance. Section 2.4 describes previous work on strongly supervised MIC, ending with a discussion on the potential issues. Section 2.5 discusses the pros and cons of working with strongly and weakly labeled data. Finally, Section 2.6 lists the three research questions that are addressed in this thesis.

In Chapter 3, the focus is on 2.6. The chapter begins with a discussion of the potential issues with applying methods geared towards strongly labeled data to weakly labeled data. Section 3.1 discusses literature in sound event detection pertaining to weakly labeled audio classification. The following Section 3.2 discusses the multi-instance learning framework and frames the MIC problem within it. An attention-based model is presented and evaluated in sections 3.3 and 3.3.2.

Generative modeling-based semi-supervised learning is discussed in Chapter 4. This

chapter focuses on 2.6 and 2.6. First, a background on semi-supervised learning methods is given in Section 4.1.1. Next, the multi-label MIC task is converted into instrument-wise binary classification problems in 4.2. Section 4.3 presents the theory behind the deep generative model, followed by the model architecture in Section 4.4, and the evaluation in Section 4.5.

Chapter 5 addresses 2.6 and studies the impact of unlabeled data in the MIC task. As in previous chapter, this chapter opens with a discussion about consistency regularization-based SSL research. Section 5.3 describes the specific version of consistency regularization used in this thesis: the mean teacher approach. The mean teacher approach is evaluated for the MIC task in Section 5.4. Section 5.5 focuses on studying the effectiveness of the method using a larger scale dataset: million song dataset for the task of music tagging.

Finally, Chapter 6 collects the various insights gained during the course of thesis. The various contributions are listed in Section 6.1. Section 6.2 closes out the thesis with a few recommendations for future research directions for music classification tasks generally, and MIC in particular.

CHAPTER 2
MUSICAL INSTRUMENT CLASSIFICATION

Musical instruments play a vital role in shaping the overall timbre or style of music. While the human ear is fairly successful at discerning and identifying various instruments being used in a song, computer algorithms are unable to perform the same task reliably. Recognizing musical instruments in an audio signal has been an active area of research in the field of MIR for the last two and a half decades [14, 64] and there has been tremendous progress in that time.

This chapter serves as an introduction to the MIC task discussing past literature for different variants of MIC. Next, various datasets that are commonly used for MIC are presented followed by metrics that are used to compare and evaluated MIC systems. A deep neural network-based method for strongly supervised MIC is presented, culminating in a discussion of various issues with the experiment setup and data. Having pointed out the issues pertaining to data, strongly and weakly labeled datasets are compared. The strongly supervised MIC experiment serves as the basis for my research questions, which are subsequently presented.

2.1 Background

A musical instrument is a specifically constructed device which when excited by force produces vibrations with distinct frequencies and amplitudes [65]. The source-filter model has also been used to describe the sound generation mechanism of instruments. In this model the musical instrument may be considered as two parts: (i) 1. the source, or the part that starts vibrating (e.g. strings on a guitar, drum head), and 2. the filter, or the part that resonates and shapes the tone of the resulting sound (e.g. the body of a guitar). Both of these physical entities vary across different instruments which give each instrument

its characteristic timbre. Additionally, electrical and electronic components may also be incorporated in instruments for different purposes, such as amplifying a signal in an electric guitar, or generating sound waves with various characteristics in synthesizers. As discussed in 1, the perception of timbre is closely related to spectro-temporal characteristics of the sound. Early research in MIC aimed to identify the features that made it possible for humans to differentiate and identify instruments, and subsequently also attempt to train machines to do the same [16].

The MIC task may be generally defined as follows: given a set of musical instrument labels, detect the presence, activation strength of absence of these instruments in a given audio signal. The input audio signal may be monotimbral (containing a single instrument) or polytimbral (containing multiple instruments). The musical instruments may be monophonic or polyphonic based on the ability to produce only a single note or multiple notes at the same time. In literature, the terms monophonic and monotimbral are often used interchangeably. Similarly, the terms polyphonic, polytimbral and multi-instrument are used interchangeably.

Research in MIC can broadly be divided into three distinct tasks based on the type of audio signal being analyzed:

(i) MIC with monophonic or monotimbral music,

(ii) Predominant instrument recognition in polytimbral music,

(iii) Classification of multiple instruments in polytimbral music.

Figure 2.1 illustrates the differences between the three tasks. Note that these tasks are arranged also in increasing order of complexity. The difficulty in recognizing instruments in a multi-instrument setting may be attributed to:

(i) large variance in timbre, performance style or playing techniques within the same instrument class,

(ii) perceptual similarity of different instrument classes leading to confusion,

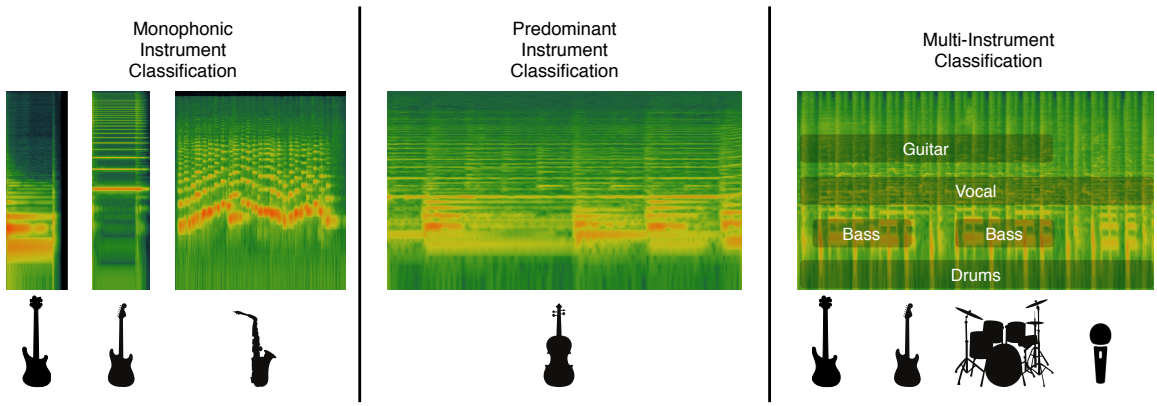

Figure 2.1: Three variants of MIC are shown above, arranged in increasing order of complexity from left to right. In the monophonic instrument classification task (left) the input audio (shown using the spectrogram representation) consists of a single instrument. The first two clips are single-note recordings. The third clip is a continuous recording of a monophonic instrument. Predominant instrument classification (center) deals with polyphonic music which may contain multiple sources. The goal of systems for predominant instrument classification is to classify the single most salient instrument in the clip. In multi-instrument classification (right) the input is polyphonic music and may contain multiple instruments. However, as opposed to predominant instrument classification, in this case all instruments in the clip are detected or classified.

(iii) superposition of instruments in time and frequency, and

(iv) masking of instruments due to different mixing levels.

Note that any mention of MIC without a qualifying prefix refers to the classification of multiple instruments, as that is the focus of this thesis.

2.1.1 MIC with Monophonic Music

In this task, instrument classification may be performed on sounds at the note-level or on continuous audio signals of solo instrument performances. This task is setup as a multi-class classification problem. This is one of the few tasks in MIR that is widely regarded as a solved problem although there are yet a few unanswered research questions [66]. An extensive survey of early work in note-level instrument classification was conducted by Herrera et al. [56]. More recent literature is reviewed in [66] for both note-level and solo instrument classification. Most research involves the traditional approach of engineering features that

reflect timbral characteristics of sound followed by ML classifiers such as Gaussian mixture models (GMMs) [57, 16] or support vector machines (SVMs) [67].

In addition to extracting spectro-temporal audio features, feature learning has also been applied to this task. Yu et al. utilize sparse cepstral codes and an SVM for classifying single-source and multi-source (polyphonic) audio [68]. Han et al. propose to use sparse coding for learning features from mel-spectrograms extracted from a dataset of single-note audio clips for 24 instruments. An SVM is trained to classify the instruments using the learned features [69].

2.1.2 Predominant Instrument Recognition

This task is an extended form of solo instrument classification and involves identifying the 'predominant' or lead instrument in a polytimbral audio signal. This task is also setup as a multi-class classification problem. The typical setup for predominant instrument classification involves model training using short clips of polytimbral audio labeled with a single predominant instrument. During test time or inference time, audio is segmented into short clips (of equal length as training clips) and passed to the system for prediction. This circumvents the potential issue of there being multiple predominant instruments.

Eggink and Brown utilize pitch tracking and extract the partials of the most dominant fundamental frequency to identify a solo instrument in polyphonic backgrounds achieving 86% accuracy on 5 instruments [70]. Livshin and Rodet used a feature selection method with a large set of audio features to perform close to real-time solo instrument classification [71]. They achieve 85% accuracy on 7 instruments with the reduced feature set.

Fuhrmann et al. extract a large set of features representing a frame and perform predominant instrument detection in real-world audio signals using one SVM per instrument [72]. Bosch et al. extend the work by utilizing source separation to segregate the polyphonic audio into streams: 'bass', 'drums', 'melody' and 'other'. The segregated audio is subsequently used for classification [73].

Han et al. apply deep convolutional neural networks (CNNs) for the task and report a significant improvement of results over previous approaches [18]. They use sliding windows of short clips of audio during test time and aggregate the model predictions using various post-processing techniques. They achieve a macro-average F1-score of 50.3% on 11 instruments.

2.1.3 Multiple Instrument Classification

This task further extends predominant instrument classification to also identify background instruments. Usually, methods for this task aim to detect the presence or absence of a predetermined set of instruments in input audio. This task is setup as a multi-label classification problem. The major difference between this task and the predominant instrument recognition task lies in the fact that predominant instrument recognitions methods only need to recognize the foreground instrument, which is often louder and therefore easier to classify than background instruments.

Kitahara et al. extract spectral and temporal features along with PCA and Latent Discriminant Analysis (LDA) for classification of five instruments in duo, trio, and quartet music [74]. The data is generated using instrument sound samples and MIDI files from the RWC dataset [75]. The desired number of instruments is obtained by mixing different voices together. Heittola et al. combine the results of Non-negative Matrix Factorization (NMF) with excitations of notes obtained from a multi-pitch tracking algorithm [76] to extract harmonic spectra from a mixture signal. The separated spectra are represented by MFCCs and classified with a GMM [77]. They also use the RWC dataset [75] to generate audio data with up-to 6 note polyphony, achieving 59% recognition rate on 19 instruments.

Recently, multi-track recordings have gained popularity in the multi-instrument classification task. The benefit of using multi-track data is that fine-grained instrument activity labels can be automatically obtained. Section 2.2 discusses more details about multi-track datasets for MIC. Li et al. [22] trained CNNs on raw audio for multi-instrument classifica-

tion using the MedleyDB dataset [78]. Gururani collected, processed, and released a new multi-track dataset called Mixing Secrets [79] which was then combined with MedleyDB and utilized for multi-instrument classification [20]. Section 2.4 discusses my experiments with multi-track data in more detail. Hung et al. utilized the fine-grained (audio frame-level) instrument activity and pitch annotations in the classical music dataset: MusicNet [48] and showed the benefits of pitch-conditioning on model performance [19]. Hung et al. further showed that a multi-task learning approach for instrument recognition, where instrumentation and pitch are jointly predicted, outperformed pitch-conditioned models [55].

2.2 Datasets for MIC

There have been several datasets made available for instrument recognition. Each is suitable for different forms of instrument classification such as the MUMS dataset [80], UIowa MIS [81], RWC dataset [75] for monophonic instrument classification, IRMAS dataset [73] for predominant instrument classification, and the OpenMIC dataset [82] for MIC. Here the focus is on datasets for polytimbral MIC.

Datasets for MIC can be categorized as:

- Strongly Labeled Dataset (SLD): clips of audio are annotated with the precise timing of presence and absence of instrument activity [83].
- Weakly Labeled Dataset (WLD): presence or absence or instruments is indicated in audio clips. Richer annotations, such as, timing, or duration of relevant instruments is not available [83].

Note that this definition of 'weak labels' is loosely connected to the corresponding usage of the term in image classification. Weakly labeled images refer to images that are collected using keyword-based image search engines [84]. Compared to manually labeled images, weakly labeled ones are noisy in nature and lead to poor classification performance of supervised learning-based approaches.

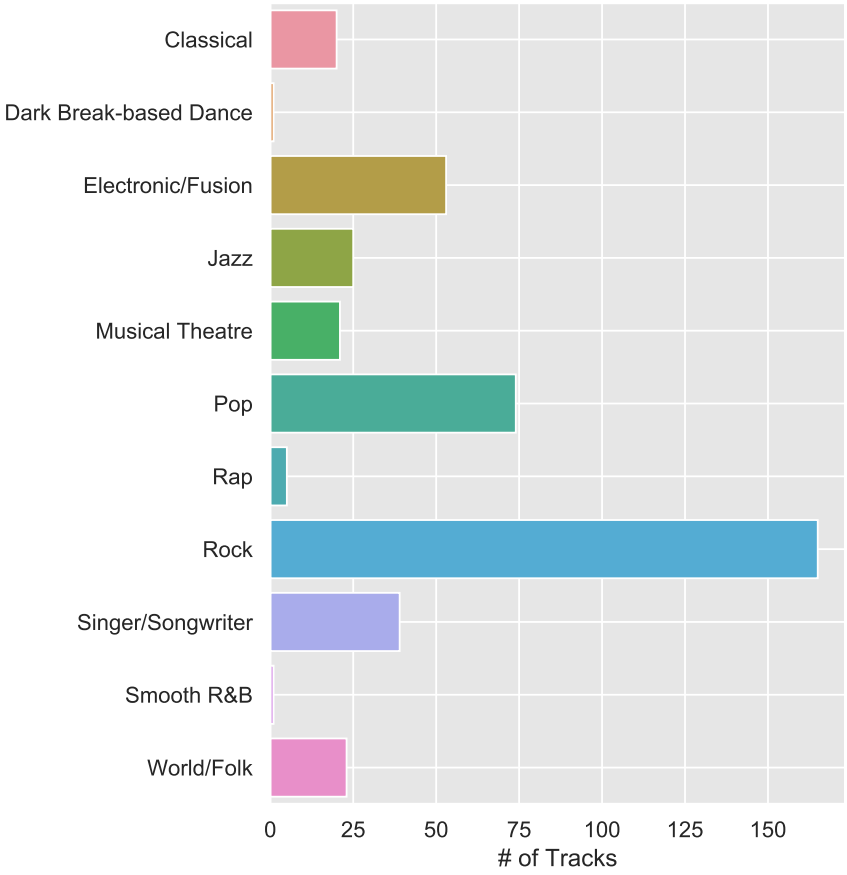

Figure 2.2: The genre distribution of the combined MedleyDB and Mixing Secrets dataset after accounting for bleed. Note that the genre labels are chosen based on those available in the MedleyDB dataset. For the Mixing Secrets dataset, the genre labels included sub-genres such as technical death metal. The sub-genres were mapped to the closest MedleyDB genre.

Most past research for MIC primarily focuses on utilizing SLDs for training. A challenge with building datasets — especially SLDs — for MIC is that it is difficult as well as expensive to completely label *all* time-steps of a song with *all* the instruments present in it. One way to address this is to synthesize polyphonic, multi-instrument music using single-note data and MIDI files [85, 74, 77]. Recent studies [19, 20, 22] utilized multi-track datasets such as MedleyDB [78] and Mixing Secrets [79] to obtain fine-grained annotation for instrument activity from isolated instrument stems or tracks.

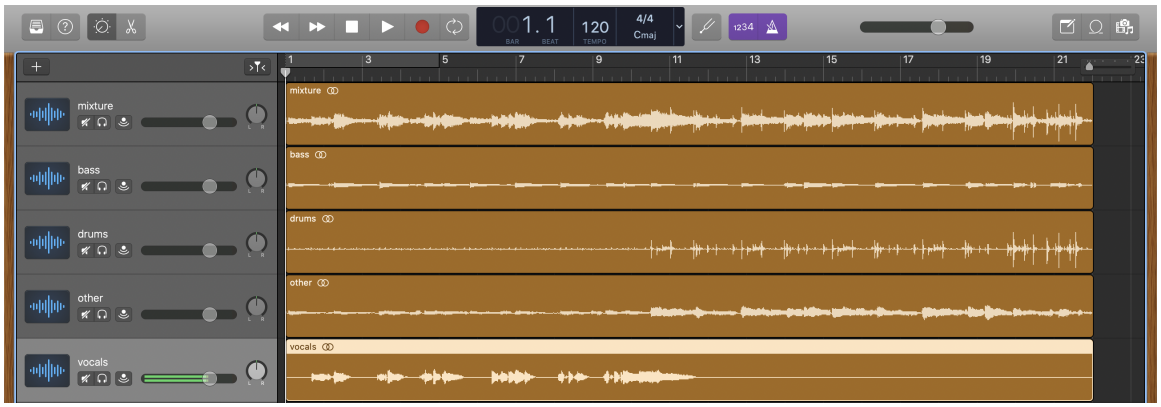

Figure 2.3: Multi-Track data in a Desktop Audio Workstation (DAW). The 'mixture' track is a mix of the four other tracks, also known as 'stems', with varying mixing coefficients. These tracks are collected during a recording session. The instrument activity may be estimated by processing the stems.

2.2.1 Multi-track Datasets

Multi-track data are a rich source of data for MIR tasks. These usually contain complete recording sessions for songs which in turn are composed of: (i) raw tracks, which are usually raw recordings of individual instruments, (ii) stems, which are post-processed versions of the raw tracks, and (iii) mix track, which is the final mixed version of the song. Multi-track data is appropriate for MIC since the instrument stems can be processed to obtain the instrument activations in the mix track at any desired time-resolution. Thus, SLDs for MIC may be constructed using multi-track data. An illustration of multi-track data is shown in Figure 2.3.

The MedleyDB [78] and Mixing Secrets datasets [79] are two multi-track datasets that have been utilized for MIC. MedleyDB consists of 254 publicly available multi-tracks and 76 additional multi-tracks available upon request. The Mixing Secrets dataset consists of 258 multi-tracks. One issue with multi-track data is that there may be tracks where the stems have crosstalk or *bleed*. For such tracks, stem activations for a certain instrument may contain activity from another instrument. Such tracks are ignored for the purpose of MIC. The genre distribution for the combined dataset is shown in Figure 2.2.

2.2.2 OpenMIC Dataset

The OpenMIC is a recent effort by Humphrey et al. to create a large scale open licensed music dataset for MIC [82]. The dataset is the largest scale WLD for MIC, containing 20000 10 s clips, each from a unique song, and annotated with up to 20 instrument labels. The songs are chosen from the FMA dataset [86], a large collection of freely available music. A property unique to the OpenMIC dataset is that it contains both positive and negative classes. This implies that clips with positive class of an instrument contain that instrument, while those with the negative class of an instrument do not contain that instrument. Clips without a positive or negative class label for an instrument may or may not contain that instrument. Thus the OpenMIC dataset may be classified as a partially labeled dataset with missing labels.

2.3 Evaluation Metrics

2.3.1 Monophonic MIC

The task of monophonic MIC is formulated as a multi-class classification problem. Systems are required to pick one of the instruments they have been trained to classify. Most literature in monophonic MIC utilizes standard metrics for multi-class classification for evaluation such as accuracy or error rate, precision, recall, and F1-score or F-measure. The report aggregate metrics, micro-average and macro-average strategies are used.

$$\text{micro-average accuracy} = \frac{C}{N}, \tag{2.1}$$

where C counts all correct classifications. N is the total number of data points.

$$\text{macro-average accuracy} = \frac{\sum_{i=1}^{M} \frac{C_i}{N_i}}{M}, \tag{2.2}$$

where M is the number of instruments being classified, C_i is the number of correct classifications for instrument i, and N_i is the total number of data points for instrument i. The micro-average strategy is used for balanced datasets while the macro-average strategy is used for imbalanced datasets to avoid skewing results for minority instruments.

2.3.2 Polytimbral MIC

Evaluating polytimbral MIC systems is not as straightforward as monophonic MIC. MIC is a multi-label classification problem, meaning that each data point may be associated with multiple instrument labels. These labels may be of a positive or negative class, indicating the presence or absence of the corresponding instrument label, respectively. Thus, evaluation is performed by treating MIC as multiple binary classification problems.

Standard binary classification metrics are used to evaluate the performance per instrument label, such as accuracy, precision, recall, and F1-score. These metrics require a threshold to be used to make a classification decision. To compare methods, threshold agnostic binary classification metrics such as area under receiver-operator characteristic curve (AUROC) is also used. While the AUROC is a valid metric for comparison of systems, especially for imbalanced classes where thresholding might be tricky, metrics such as accuracy, precision, recall, and F1-score paint a clearer picture about the actual performance of individual systems.

For the methods described in this thesis I used the following evaluation methodology. Since the MIC problem is typically imbalanced, it is not appropriate to use simple binary classification accuracy. Adhering to best practices for imbalanced binary classification, the precision, recall and F1-score to evaluate performance of all models per instrument and macro-averaged. Note that precision, recall and F1-score ignore the true negative predictions. This is not ideal since the data can be imbalanced between positives and negatives. For a fair evaluation, the metrics are computed for both classes separately. The F1-scores for both positive and negative class are averaged to obtain a macro-averaged F1-score per instrument.

Note that in terms of binary classification, the F1-score for negative class is simply the harmonic mean between the negative predictive value and the specificity. This methodology is a modification of that utilized by Humphrey et al. [82] to evaluate their baseline model on the OpenMIC dataset. They utilize the support weighted average of the two F1-scores instead of macro-average. This skews the metric in favor of the majority class for each instrument label.

Finally, to compare the overall performance of different models with each other, we average the precision, recall, and F1-scores of each instrument to obtain the overall macro-averaged precision, recall, and F1-score, which will be referred to as overall precision, recall, and F1-score respectively.

2.4 Strongly Supervised Instrument Classification

This section focuses on some of my earlier work on multi-instrument classification [20]. The term 'strongly supervised' has to do with the nature of data being utilized to train models for MIC. *Strong supervision* in the context of this thesis implies that MIC models are trained using SLDs. Similarly, *weakly supervised* MIC implies that a WLD is utilized instead.

For this experiment, the task is to train deep neural networks using strongly labeled data to detect the presence of 18 instruments in polytimbral music. We named this task instrument activity detection (IAD) as it deals with fine time resolution (1 s) which may be used to obtain time-series of instrument activity in longer clips of audio. Multi-track datasets MedleyDB and Mixing Secrets are combined for this experiment. The models are trained using 1 s clips of audio annotated with instrument activity. We compared simple multi-layer perceptrons (MLP), CNNs, and convolutional recurrrent neural networks (CRNNs) for this task.

2.4.1 Experiment

The premise of this experiment was to determine the effectiveness of different kinds of neural network architectures for the IAD task. DNNs are particularly useful due to their ability to approximate complex non-linear functions mapping an input feature space to the outputs. Thus, raw or minimally processed data can be used as an input to a DNN which may then learn features relevant for the task at hand. We utilize three standard architectures for the task: MLP, CNN, and CRNN. The convolutional layers in the CNN and CRNN incorporate inductive biases shown to work well with audio spectrograms [87, 88]. Furthermore, the recurrent layer in the CRNN incorporates the ability to learn longer term temporal features and has shown improved performance over CNNs in tasks such as music tagging [31] and sound event detection [89].

Pre-processing

First, all mix tracks are split into 1 s clips and the corresponding instrument activations are obtained. The raw audio data is transformed into a magnitude spectrogram using window and hop size of 46.4 ms and 11.6 ms, respectively. The audio sample rate is 44100 Hz. The linear frequency and magnitude axes are converted to logarithmic using 96 mel filters and decibel scaling, respectively. The mel filters are triangular overlapping filters which apply a non-linear transformation to the frequency scale to mimic human perception of frequency. Thus, we obtain a log mel-spectrogram of dimensionality 96×86 as the input for our neural networks. Additionally, the training data is augmented using pitch shifting (6 semitones up and 5 semitones down, with 1 semitone increments).

Model Architectures

- **Multi-Layer Perceptron (MLP)**: A simple architecture is chosen for the MLP or fully connected model: 4 hidden layers with 256 hidden units in each layer. Dropout [90] is used as a regularizer with a keep probability of 0.5 at each layer.

Table 2.1: Model Architecture. (Conv2D: 2D Convolutional Layer, MP: 2D Max-Pooling, k: kernel size, d: filter depth)

CNN	**CRNN**
Conv2D $k = 3 \times 3, d = 64$ MP $(p = 2, 2)$	
Conv2D $k = 3 \times 3, d = 128$ MP $(p = 2, 2)$	
Conv2D $k = 3 \times 3, d = 256$ MP $(p = 3, 3)$	Conv2D $k = 3 \times 3, d = 256$ MP $(p = 2, 2)$
Conv2D $(k = 3 \times 3, d = 640)$ MP $(p = 3, 3)$	Conv2D $(k = 3 \times 3, d = 256)$ MP $(p = 2, 2)$
FC $(h = 128)$	GRU $(h = 256)$
FC $(h = 18)$	

Note that the log mel-spectrogram input is flattened when used with this network.

- **Convolutional Neural Network (CNN)**: The CNN architecture is shown in table 2.1 (left). All convolutional filters are 3×3, with a stride of 1 and zero-padding of 1 (to preserve input size during the convolution operation). Each Conv2D layer is followed by batch-normalization [91] and the Exponential Linear Unit (ELU) [92] activation function. We use max-pooling after each convolutional block as well. The final convolution layer's output is flattened before passing it to a fully connected layer, followed by ELU activation. Finally, we connect to an output layer of 18 units with a sigmoid activation function.

- **Convolutional Recurrent Neural Network (CRNN)**: The CRNN architecture is shown in table 2.1 (right). The architecture is similar to the CNN model, with a few hyper-parameters changed. We utilize the gated recurrent unit (GRU) in the recurrent network. For the recurrent network to function properly, we preserve the time dimension of the output while flattening the final convolutional layer's output. Thus, only the depth and height dimensions are flattened. Finally, the last GRU output is connected to the output layer consisting of 18 units with a sigmoid activation

Table 2.2: Multi-track dataset instrument distribution

		Train		Test	
Instrument	Abbr.	# Tracks	# 1 s clips	# Tracks	# 1 s clips
drum set	dru	300	720036	79	15957
electric bass	bgtr	253	620592	62	13344
male singer	ms	200	351384	62	10038
dist. elec. gtr	dgtr	171	396204	40	7522
clean elec. gtr	cgtr	119	225456	34	5875
synthesizer	syn	118	295524	33	5712
acoustic gtr	agtr	91	230556	25	5241
piano	pf	89	187536	24	4063
vocalists	vox	84	154596	12	1895
female singer	fs	79	149232	23	3733
string section	str	24	39444	10	1278
elec. piano	epf	24	52680	14	2075
elect. organ	eorg	22	39516	11	2117
double bass	db	21	40116	9	1786
cello	vc	13	22176	9	1623
violin	vn	10	28452	15	2385
tabla	tab	9	41640	3	806
flute	fl	7	9972	7	1171

function.

Training Procedure

Binary cross-entropy is used as the loss function for all models. Stochastic gradient descent with a learning rate of 0.0001 and momentum of 0.9 is used to optimize the loss function. The models are trained using mini-batches of 32 instances for 20 epochs, which is sufficient for the training and validation loss to converge for each of the architectures.

Evaluation

The combined MedleyDB and Mixing Secrets datasets consist of 461 tracks after removing tracks with bleed. These tracks are divided into a training and test set such that they do not share any tracks by the same artist. The class distribution for the training and testing set is shown in table 2.2.

As is evident from table 2.2, there is severe imbalance in the dataset. Therefore, for a fair comparison, the metrics used to evaluate the IAD models should not be influenced by

Table 2.3: Mean AUROC for the three models across various evaluation time-resolutions

	MLP	CNN	CRNN
1 s	71.28	**77.55**	77.5
5 s	70.85	78.35	**78.76**
10 s	70.82	78.59	**79.22**
Track	71.1	**80.92**	80.1

imbalance. Additionally, as discussed in Han et al. [18], binarizing model outputs using a fixed threshold and evaluating the accuracy depends on the selected threshold. For this experiment, a threshold agnostic metric: AUROC is used for evaluation. The AUROC is computed by first plotting the true positive rate and false positive rate on a plane for various classification thresholds, which results in a curve. AUROC is the area under this curve. It measures the probability that the model assigns a higher score to a randomly selected positive instance than a negative instance. Since AUROC is usually applied to binary classification, we compute it per instrument class. We report an average AUROC by taking the mean of the AUROC per class.

Since the neural networks are trained using 1 s snippets of audio, a prediction is made for every 1 s in the test track. We evaluate the models at varying time-resolutions by post-processing the predictions using max-pooling operations according to the desired time-resolution. For example, in order to have a 5 s second resolution, the maximum across 5 continuous predictions for every instrument is chosen as the predicted score for the corresponding 5 s clip in the track.

Results

A comparison of model performance is summarized in table 2.3. It can be observed that CNN and CRNN outperform MLP in both metrics. This is expected since the convolutional layers allow the model to learn hierarchical acoustic features from the time-frequency representation more efficiently. However, the CRNN does not outperform the CNN, which may be attributed to the fact that only 1 s snippets are used. The benefits of using recurrent layers should be more noticeable when longer sequences are involved. In our experiments,

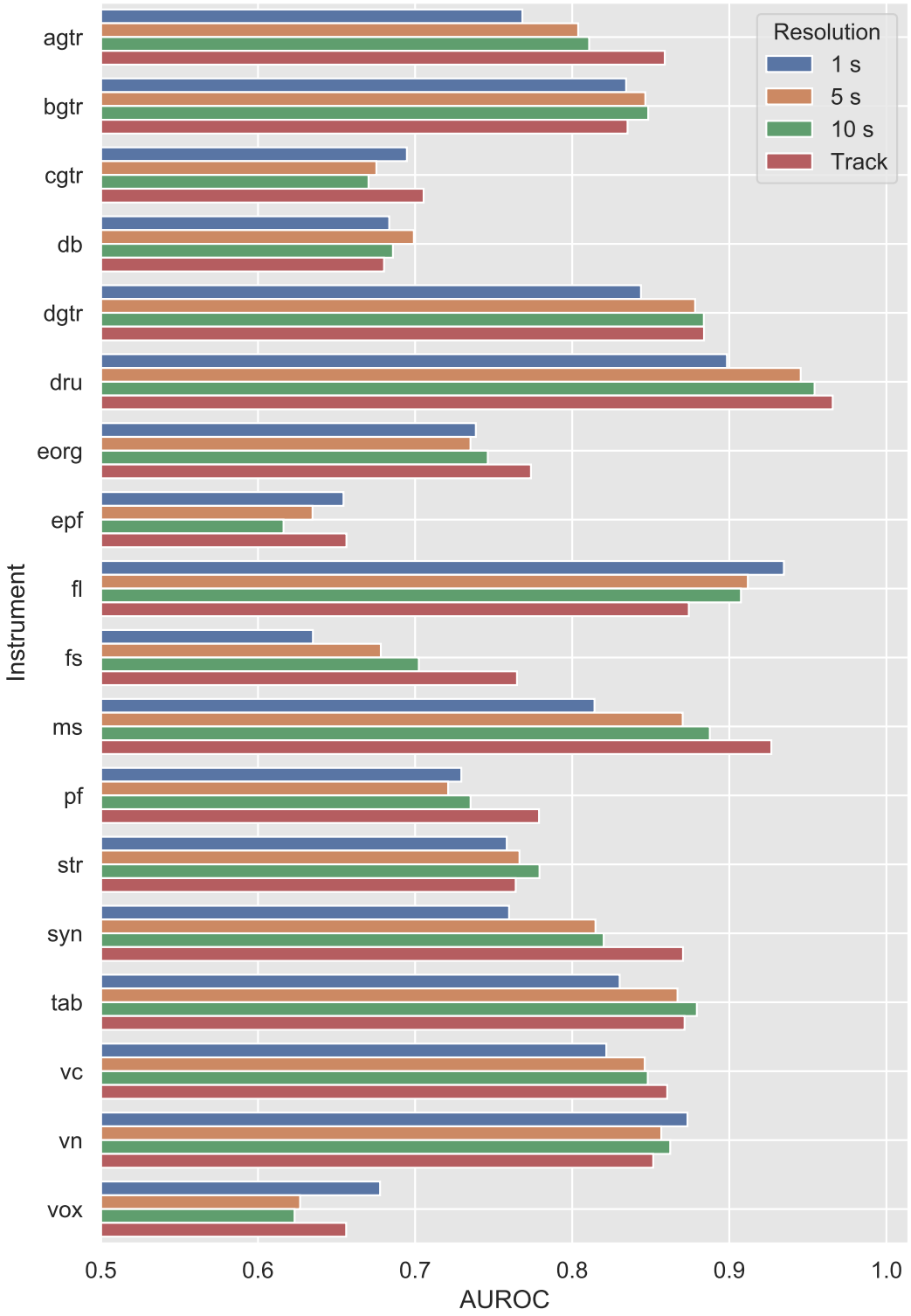

Figure 2.4: AUROC per instrument for CRNN model. The results are shown for evaluation performed at different time resolutions.

the temporal dimension of the input is reduced to only 5 time steps after the 4 CNN layers. Another observation is that output aggregation tends to improve models' label ranking performance. A similar trend is observed for mean AUROC.

Figure 2.4 shows the AUROC per instrument of the CRNN model for the chosen time-resolution aggregation. Some noticeable observations are as follows:

- Using output aggregation in time leads to better performance in almost all instrument classes. This is expected.
- The model achieves high AUROC not only for majority instruments in the dataset but also for minority instruments such as flute, violin and cello suggesting that the model is able to detect instruments correctly to some degree and is not simply predicting the majority instruments.
- An additional experiment was carried out to investigate the classification thresholds for different instruments. It was found that the best threshold chosen per instrument for binarization ranges from 0.02 to 0.55 with lower thresholds for minority instruments in general. We observe a correlation coefficient of 0.9 between the thresholds and the training data distribution suggesting that the model has learned biases in the dataset.

2.4.2 Discussion

The purpose of this experiment was to evaluate the effectiveness of different neural network architectures for the IAD task. Towards that end, it was established that CNNs and CRNNs are indeed effective for instrument classification — reinforcing past music classification literature. However, there are a few issues that were revealed through this experiment from the perspective of making practical MIC systems.

First, a glance at table 2.2 reveals the degree to which the dataset is imbalanced. Similarly, the genre representation is not balanced, with Rock and Pop music (see Figure 2.2) dominating all other genres. Interestingly, the unprocessed MedleyDB consists of a large

number of classical music tracks but a vast majority number of these are not included in the final dataset since the stems for these tracks have *bleed*. Note that imbalanced data is not necessarily a bad thing since there may be an implicit bias in the distribution of different classes which should indeed be modeled. However, this particular dataset is constructed using only 450 unique music tracks, with multiple tracks from the same artist. This is a very small number of unique songs and artists to train generalizable models for MIC using this dataset.

2.5 From Strongly to Weakly Labeled Data

In Section 2.1.3, a few difficulties in developing MIC systems are listed. These challenges are inherent to the instrument classification task itself and have to do with the properties of the audio. There is, however, an additional challenge, which I discuss in sections 1.3 and 2.4.2. Most state-of-the-art methods, and the methods discussed in this thesis, are based on data-hungry deep neural networks (DNNs) which are sensitive to the scale and quality of data being used to train them. There is a need for larger-scale and diverse data for the task of MIC. However, curating and annotating large-scale and diverse labeled datasets manually is infeasible. The currently popular, yet small and imbalanced multi-track SLDs are not ideal for training reliable MIC models. Therefore, larger WLDs can potentially circumvent the challenge of scale and diversity.

SLDs require expert listeners to annotate instrument activity at a fine time resolution which is usually expensive as well as time consuming if done for large collections of music. WLDs, on the other hand, are easier to collect since only the presence or absence of an instrument is required to be annotated [82]. The pros and cons of using WLD versus SLD shall be discussed in more detail in Section 2.5.1, but the core challenges lie in the fact that methods utilizing WLD have significantly fewer supervisory signals compared to those trained with SLD. This makes it challenging to simply apply existing algorithms which are designed for SLDs to WLDs for MIC.

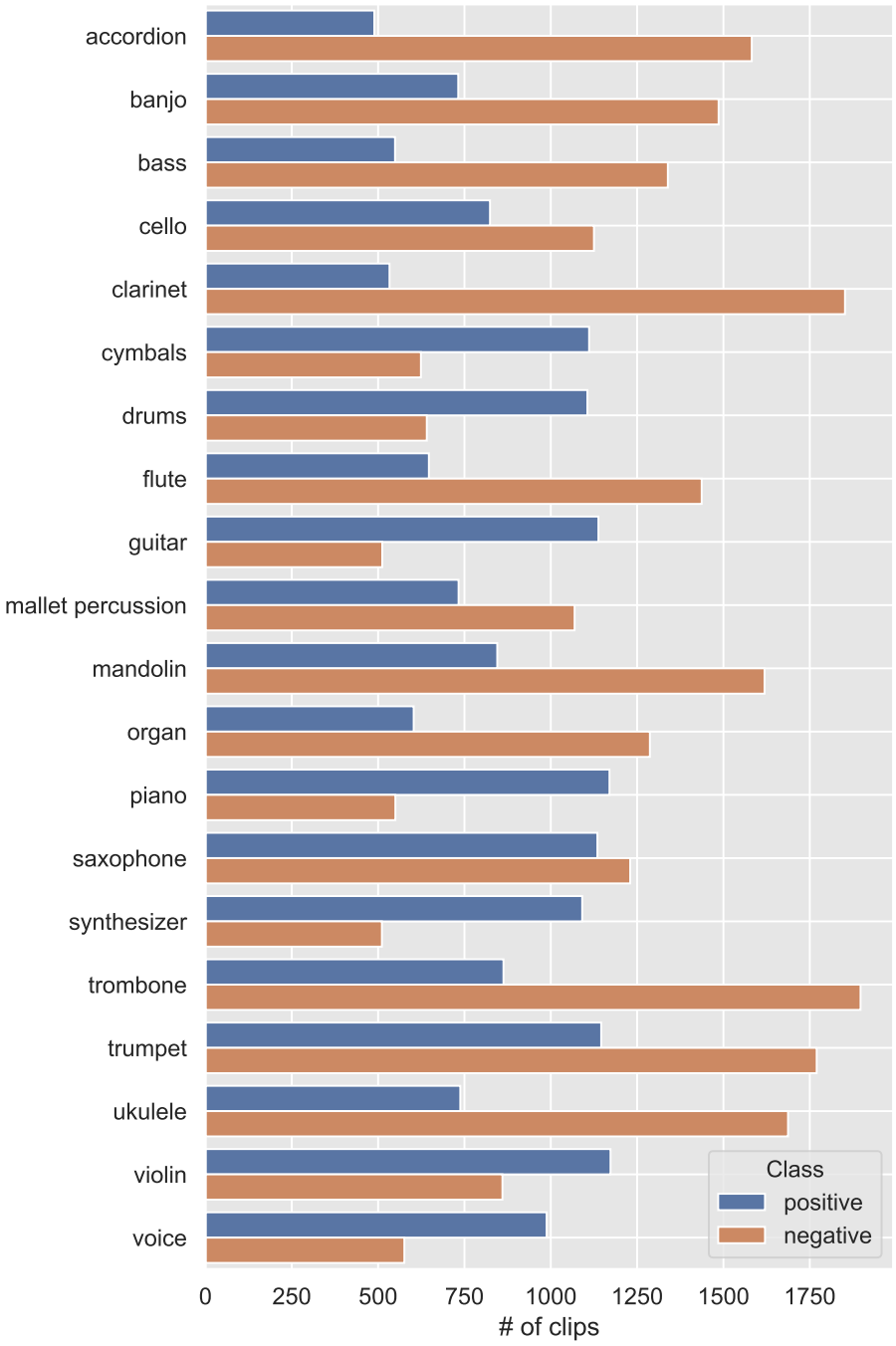

Figure 2.5: Instrument distribution for the OpenMIC dataset

The issue with these SLDs are two-fold:

1. Since they contain a few hundred songs they lack the variety of musical genres, artists, or styles for the classifiers trained to be generally useful.

2. The distribution of instruments is highly skewed.

Both of these issues were targeted by Humphrey et al. while creating the OpenMIC dataset. The genre distribution of the OpenMIC dataset is better balanced compared to that of the multi-track datasets. The diversity and scale of the OpenMIC dataset (approximately 55 hours) makes it suitable for deep learning techniques. Each clip has a set of positive labels and negative labels associated with it. A clip contains the positively labeled instruments and does not contain the negatively labeled instruments. A positively labeled instrument implies that the instrument is active *at least once* in the 10 s clip, implying **weak labels**. Also note that the instruments not occurring in either the positive or the negative instruments set *may or may not be present* in the clip, implying **missing labels**. All of the methods presented in the thesis are evaluated using the OpenMIC dataset. The instrument distribution of the OpenMIC dataset is shown in Figure 2.5. From the statistics, it is clear that while the total number of samples (positive + negative) for each instrument are approximately balanced there is an imbalance between positive and negative samples of a few instruments. This imbalance poses an additional challenge but is not specifically the focus of the methods presented in this thesis.

2.5.1 Data Challenges Revisited

Section 1.3 discussed data challenges that are shared within the MIR community. Section 2.4.2 discusses issues that pertain to SLDs specifically being used for MIC research. The OpenMIC dataset resolves most of the issues pointed out in these discussions — specifically targeting data scale, music diversity, and label balance. However, current methods for MIC need to be adapted due to the weakly labeled nature of the OpenMIC dataset. The

following new data challenges are posed by the introduction of the weakly labeled OpenMIC dataset:

i Models need to be trained using long clips of weakly labeled audio. Methods for MIC which are trained using SLDs rely on labels at the frame-level or at 1 s time resolutions. The IRMAS dataset, which is also weakly labeled, uses only 3 s clips which are considerable shorter than OpenMIC's 10 s clips.

ii Models need to be trained using partially labeled data. While there are 20,000 audio clips, each clip is not labeled with the presence or absence of all 20 instruments.

2.6 Research Questions

Having explored strongly supervised MIC and uncovering various issues (see Section 2.4.2), the primary goal for this thesis, as touched upon briefly in Section 1, is to develop DNN-based methods for MIC that are able to leverage relatively large scale WLD. Such methods fall under the category of weakly supervised learning. In the MIC task, supervision in the learning process is 'weak' either due to data being weakly labeled, or unlabeled data being utilized along with labeled data. The specific research questions the methods in this thesis aim to answer are:

(RQ1) **What modeling techniques are effective in leveraging weak labels?**

Standard methods of classification analyze the entire input and predict class labels. Usually, these methods work well with SLD where models are trained to classify short excerpts of the input audio [20, 22]. The task is difficult with WLDs since labels are only available for longer inputs — 10 s in the case of OpenMIC dataset — and the instrument may only be active at certain points in the clip. Models need to be robust to cases where an instrument is briefly present in the entire clip. For this purpose, models that intelligently aggregate predictions may be well suited for weakly labeled MIC since they ease the problem by separating the tasks of localization and classification.

(RQ2) **How can generative models be utilized for MIC?**

The field of MIR has predominantly utilized discriminative models for various classification tasks, including instrument recognition [56, 18]. These methods are especially useful if one works with large SLDs. However, due to the nature of how musical datasets are annotated, it may be more beneficial to investigate generative modeling techniques which are able to leverage partially labeled datasets. Additionally, generative models are capable of jointly modeling the input data and available labels. This property may prove vital in improving model performance.

(RQ3) **How does the inclusion of unlabeled data impact model performance?**

RQ2 focuses specifically on generative models and whether they can be utilized for semi-supervised MIC. RQ3, more generally, is about the impact of utilizing any form of extra information in addition to the existing labeled data. The extra signals may come from model inference on missing labels, or by utilizing external unlabeled data. There are a few large collections of music available [58] albeit without relevant instrument annotations. These collections may be used to augment annotated corpora and used in semi-supervised models.

The remainder of this thesis consists of three parts, each answering at least one of the aforementioned research questions. Section 3 describes an attention-based approach to MIC using WLD. The method treats the MIC task as a multi-instance learning problem and uses a simple attention mechanism to identify the most salient instances for the instruments present and aggregates the instance-level predictions. This method is a weakly supervised discriminative model that aims to answer RQ1. Section 4 discusses a generative semi-supervised learning approach for MIC, which pertains to RQ2. Variational autoencoders (VAEs) are utilized to leverage data with missing labels in a semi-supervised deep generative model. This method aims to answer RQ2 which pertains to the effectiveness of generative models for MIC. RQ3 is also addressed with this method as it utilizes labeled and unlabeled data. Finally, a consistency-regularization approach to semi-supervised learning is discussed

in Section 5. This approach utilizes a combination of stochasticity and model weight averaging to leverage missing labels or unlabeled data. An in-depth analysis is performed to study the effect of the amount of labeled data as well. This method focuses on RQ3.

CHAPTER 3
ADDRESSING WEAKLY LABELED DATA

As discussed in Section 2.2, datasets for MIC are either strongly or weakly labeled. Most prior research in MIC utilize strongly labeled datasets (SLDs) since they are rich in information about fine-grained instrument activity. This enables methods to be trained using short clips of audio with exact instrument activity. During inference, sliding windows of audio with the same length as training clips are used to identify the instruments in the clips. Figure 3.1 depicts the standard setup of deep neural network-based MIC models which are trained using SLDs [20, 19]. It is not straightforward to apply techniques for instrument recognition as the one depicted in Figure 3.1 to WLD since exact instrument activity is unknown within a long weakly labeled clip. There are two main challenges:

- Models based on SLDs are trained with instrument activity at the 1 s or frame-level. This granularity of instrument activity annotation is absent in OpenMIC where each clip is 10 s long. A simple approach is to train the models with the entire 10 s clip as input.

- Annotations in OpenMIC are at the clip level implying that to evaluate methods for instrument recognition operating at a short time resolution, model outputs need to be aggregated to obtain clip-level predictions.

Han et al. [18] utilize mean-pooling and normalized sum strategies for aggregation of model outputs in order to obtain clip-level predictions for predominant instrument recognition. In previous research, max-pooling was investigated as a method to aggregate predictions for evaluation with different time resolutions [20]. In this chapter, the MIC task is formulated as a multi-instance learning (MIL) problem [93] and an adaptive strategy for aggregating instance-level predictions based on attention mechanisms is presented.

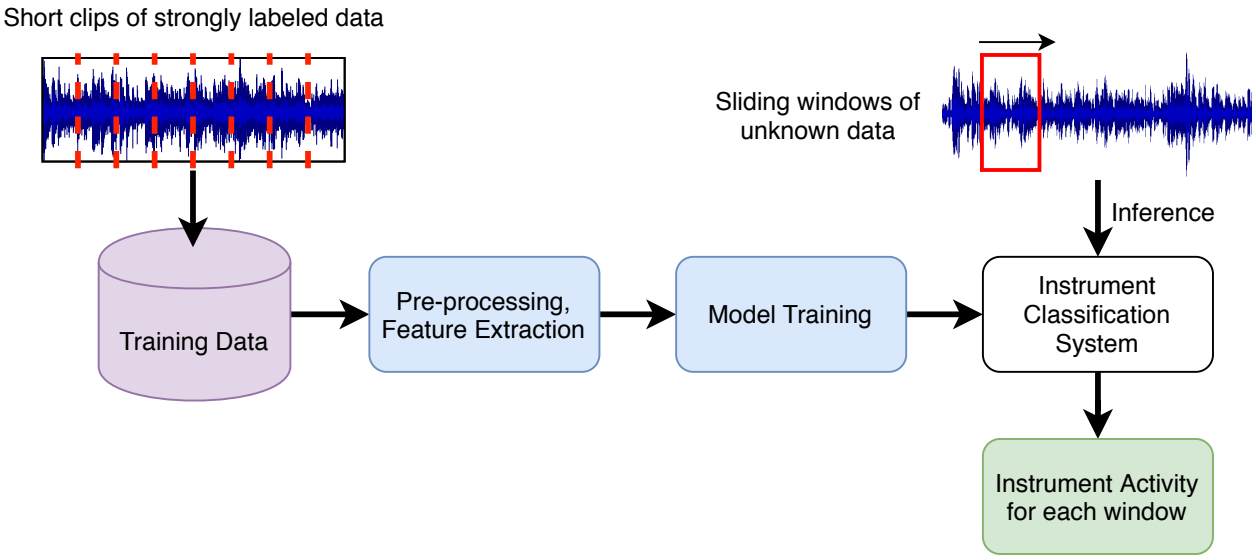

Figure 3.1: A standard approach for MIC using strongly labeled data. Purple box indicates data, blue boxes indicate processing steps, white box indicates the system, and green box indicates a system output. The training data comprises of short clips of audio along with instrument annotations. Pre-processing steps involving volume normalization, spectrogram extraction, etc are performed and features are extracted from the training data. A model is trained using this training data and evaluated using a validation dataset (not shown). Finally, during inference or testing, input audio is processed into short clips of the same length as used during model training. For this a sliding window or non-overlapping windows may be used. Finally, an estimate of the instrument activity may be obtained for the entire clip. In case the test clips are weakly labeled, the output instrument activity may be post-processed using averaging or max-pooling [18, 20].

3.1 Related Work

The task of audio or sound event classification or detection shares many commonalities with instrument recognition. Both tasks aim to identify a time-varying sound source in a mixture of multiple sound sources. One of the main differences between sound event detection and MIC stems from the nature of sound events that both tasks aim to classify. The former deals with uncorrelated sounds such as motor noise, car horns, baby cries, or dog barks, while musical audio has a rich harmonic and temporal structure often absent in audio captured from real world acoustic scenes. Stowell et al. conducted a historic review of work in sound event and audio classification [94]. This section focuses on recent literature involving deep neural network architectures —which are now the standard approach— as well as on methods that focus on addressing weak labels.

Similar to work described in Section 2.1.3, CNNs are the architecture of choice for audio classification tasks. Hershey et al. [87] adapted deep CNNs from computer vision literature and found that they are effective for large-scale audio classification. Cakir et al. [89] studied the benefits of CRNNs for sound event detection over CNNs. They found that RNNs capture long-term temporal context which helps improve performance against models only comprising CNNs.

Recent research has started to focus on using weakly labeled audio due to the reasons mentioned in Section 2.2: difficulty of annotating exact occurrences of sound events, and larger scale of available weakly labeled data. For audio classification in weakly labeled data, most research utilizes the MIL framework, where each example is a labeled bag containing multiple instances whose labels are unknown. Kumar and Raj [83] trained support vector machines (SVMs) and neural networks as bag-level classifiers capable of instance-level prediction and are hence also useful for localization of sound events in time.

With the introduction of attention mechanisms [95] some researchers adopted attention to MIL. Kong et al. [96] proposed decision-level attention to solve the MIL problem for

AudioSet [97] classification. Attention is applied to instance-level predictions to enable adaptive weighted aggregation for bag-level prediction. Kong et al. [98] extended this approach and proposed feature-level attention where instead of applying attention to the instance predictions, it is applied to the hidden layers of a neural network to construct a fixed-size embedding for the bag. Finally, a fully connected network predicts the labels for the bag using the embedding vector. McFee et al. [99] compared various methods for aggregating or pooling instance-level predictions. They developed an adaptive pooling operation capable of interpolating between common pooling operations such as mean-, max- or min-pooling.

3.2 Problem Formulation

In general, MIC can be framed as a Multi-Instance Multi-Label (MIML) classification problem [100, 101, 102]. Within the MIML framework, a training dataset is represented as $\mathbb{D} = \{(\mathbf{X}_1, \mathbf{Y}_1), \ldots (\mathbf{X}_m, \mathbf{Y}_m)\}$ where $\mathbf{X}_i$ is a bag containing r instances $\mathbf{X}_i = \{\boldsymbol{x}_{i,1}, \ldots \boldsymbol{x}_{i,r}\}$ and $\mathbf{Y}_i = [\boldsymbol{y}_{i,1}, \ldots, \boldsymbol{y}_{i,L}] \in \{0, 1\}^L$ is a label vector with L labels with $\boldsymbol{y}_{i,j} = 1$ if any of the instances in $\mathbf{X}_i$ contains label j. For clarity, the indices i used to reference a specific data point are dropped and a data point is simply represented as $(\mathbf{X}, \mathbf{Y})$.

In the case of OpenMIC data, a bag $\mathbf{X}$ refers to a 10 s audio clip. The 10 s audio clip consists of multiple instance-level features. These are features extracted from a pre-trained VGGish model for audio classification [87]. The VGGish model is a deep CNN trained for audio classification on a large-scale audio dataset — 8 million audio clips from Youtube. Details of this model can be obtained in the Appendix B. [1] For each 10 s audio clip, 10×128-dimensional features are extracted where each 128-dimensional feature vector represents 1 s (0.96 s exactly), thus leading to 10 instances per bag for the MIML problem. Note that this is also a missing label problem because for a given $(\mathbf{X}, \mathbf{Y})$, not all $\boldsymbol{y}_j$ are known or

[1] In pilot experiments, VGGish features were compared to features learned from scratch using deep CNNs. VGGish features performed better even with simpler classification models such as random forests, thus inspiring the usage of VGGish features for further experiments.

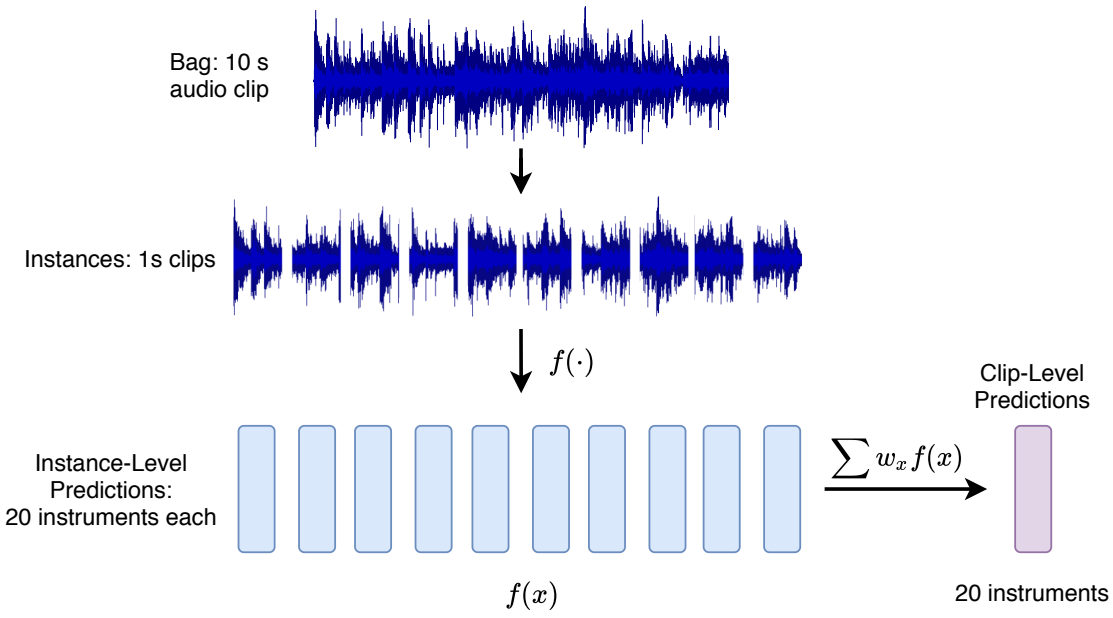

Figure 3.2: The MIML framework illustrated from the instrument classification viewpoint. A bag refers to the 10 s audio clip, which consists of 10 1 s (960 ms to be exact) instances. Each of these instances are represented as a single feature vector of 128 dimensionality. The function $f(\cdot)$ produces predictions for each instance, for each instrument. The operation $\mu(\cdot)$ performs a weighted average of the instance level predictions producing a bag-level or clip-level prediction, which may then be used for evaluation or as the inferred instrument classification output.

annotated (see Sect. 2.2.2). Figure 3.2 presents the MIL framework from the perspective of the MIC task.

In these experiments, all labels are assumed to be independent and are independently predicted for each instance. Under this assumption, the MIML problem decomposes into L (20 for OpenMIC dataset) cases of the Multi-Instance Learning (MIL) [103, 93] problem, one for each label in the dataset.

3.2.1 Multi-Instance Learning

In the MIL setting, a bag label is produced through a score function $S(\mathbf{X})$. Under the assumption of independence among instances, $S(\mathbf{X})$ can be written as:

$$S(\mathbf{X}) = \mu\Big(f(\boldsymbol{x}) \Big), \tag{3.1}$$

where $f(\boldsymbol{x})$ is a score function for an instance $\boldsymbol{x}$, and $\mu(.)$ is a permutation-invariant aggregation operation for instance scores $f(\boldsymbol{x})$ [104]. This parameterization induces a natural approach to classify a bag of instances:

(i) to produce scores for each instance in the bag using an instance-level scoring function $f(\boldsymbol{x})$, and

(ii) to aggregate the scores across different instances in the bag using the aggregation function $\mu(.)$.

Instance-Level Scoring Function $f(\boldsymbol{x})$

In this thesis I utilize two categories of neural network architectures to produce instance-level scores:

1. **Fully Connected Network**: Under the assumption of independence between instances, a simple non-linear transformation of each instance feature representation can be utilized to obtain the scores.

2. **Recurrent Neural Network**: Relaxing the assumption of independence between instances makes sense for music since neighboring instances are usually correlated. Recurrent neural networks (RNNs) are a class of NNs which are capable of learning temporal patterns in input sequences. RNN units such as bi-directional Gated Recurrent Units (GRUs) can be utilized to obtain instance-level scores conditioned on all other instances in the bag.

Aggregation Functions $\mu(.)$

Once instance-level scores are obtained, they need to be aggregated into bag-level predictions. Average or mean and Max-pooling are commonly used permutation invariant operations to obtain one prediction from several instance-level predictions. These operations however make strong assumptions about the nature of the predictions, for example, mean-pooling assumes that a majority of the instance predictions should be positive for the bag to be labeled positive. Max-pooling fits well with the paradigm of MIL but it leads to problems in terms of optimization as pointed out by McFee et al. [99]. The issue lies in the fact that during optimization non-max instances do not contribute to the gradient update of model parameters, which could be problematic at the start of training when instance scores are essentially random.

McFee et al. introduced auto-pool for adaptive aggregation of instance-level predictions [99]. The auto-pool operators address the issues of standard pooling operations providing an adaptive operator that is capable of smoothly interpolating between various pooling operations, adapting to the nature of the task. While auto-pool does behave similar to attention mechanism, the weights are learned independently of any input features, which is different from the attention mechanisms proposed for sound event detection [98] or machine translation [95]. They argue that attention mechanisms are primarily useful in tasks involving structured prediction, i.e. where the predicted output has an inherent structure (e.g. translated sentences in machine translation). Tasks such as sound event detection are unstructured and hence not appropriate for attention. Despite these arguments, empirical results [98, 96] show that attention mechanisms do tend to work well for MIL problems.

Bag-level predictions are obtained with a scoring function $\mu(.)$. This function is formu-

lated as the weighted sum of instance-level scores, i.e.:

$$S(\mathbf{X}) = \mu\Big(f(\boldsymbol{x}) \Big) \tag{3.2}$$

$$S(\mathbf{X}) = \sum_{\boldsymbol{x} \in \mathbf{X}} w_{\boldsymbol{x}} \, f(\boldsymbol{x}), \tag{3.3}$$

where $w_{\boldsymbol{x}}$ is a learnable attention weight for instance $\boldsymbol{x}$. Furthermore, instance weights w_x should sum to 1, i.e., $\sum_{\boldsymbol{x} \in \mathbf{X}} w_{\boldsymbol{x}} = 1$. This ensures that the aggregation operation is invariant to the size of the bag, thus allowing the model to work with sound clips of arbitrary length. This normalization also leads to a probability-like interpretation of the instance weights which can then be used to infer the relative contribution of each instance towards $S(\mathbf{X})$.

For an instance $\boldsymbol{x} \in \mathbf{X}$, the weight $w_{\boldsymbol{x}}$ is thus parametrized as:

$$w_{\boldsymbol{x}} = \frac{\sigma\Big(\boldsymbol{v}^\top h(\boldsymbol{x})\Big)}{\sum_{\boldsymbol{x}' \in \mathbf{X}} \sigma\Big(\boldsymbol{v}^\top h(\boldsymbol{x}')\Big)}, \tag{3.4}$$

where $h(\boldsymbol{x})$ is a learned embedding of the instance $\boldsymbol{x}$, $\boldsymbol{v}$ are the learned parameters of the attention layer, and $\sigma(.)$ is the *sigmoid* non-linearity. This attention mechanism is similar to the decision-level attention as formulated by Kong et al. [98].

3.2.2 Model Architecture

Computing the final bag-level scores $S(.)$ involves predicting instance-level scores $f(.)$ and aggregating the scores across instances using a learned set-operator $\mu(.)$ which performs weighted averaging with the weights computed with equation 3.4. For this experiment, both instance level $f(.)$ and bag-level $S(.)$ scores are represented as the probability estimate of the instance or bag being a positive sample for a given label.

First, each instance $\boldsymbol{x}$ is passed through an embedding layer of three fully connected layers to project each instance to a suitable feature space. Next, instance-level scores are computed from the output of $f(.)$ with another fully connected layer. Similarly, attention

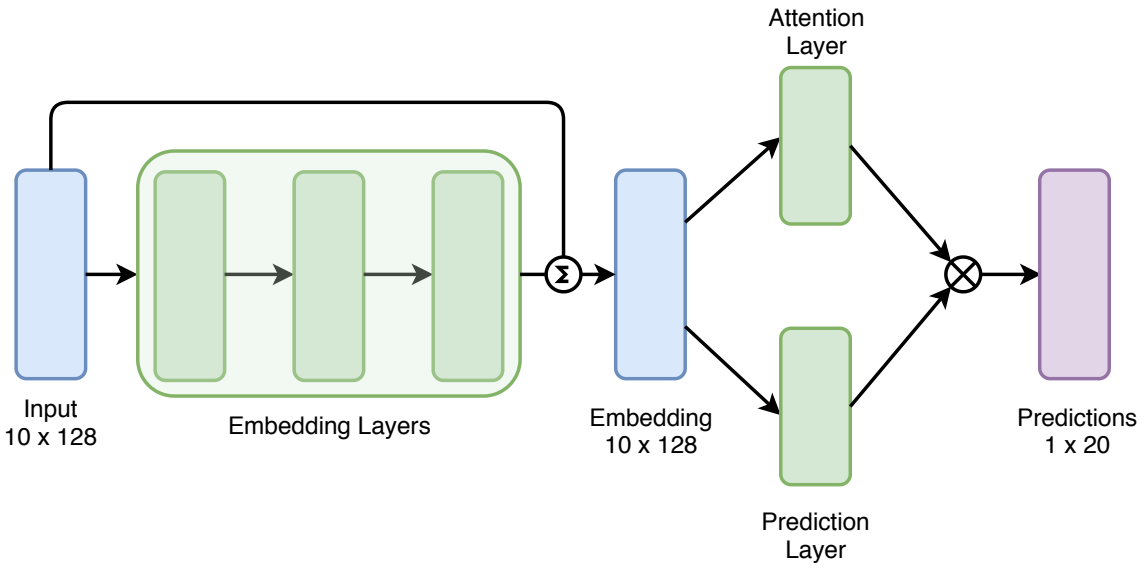

Figure 3.3: Model architecture for the ATT model. The 10 s audio clip is represented using the transfer learned VGGish features, which have a dimensionality of 10×128, where each 128-dimensional feature vector represents a single 1 s instance from the clip. These features are passed through an embedding layer. The embedding layer is a sequence of three fully connected layers which non-linearly transform each of the instance feature vectors. A skip connection from the input to the output of the embedding layers is used. The prediction layer outputs the instance-level predictions for each embedding instance. The attention layer estimates the normalized attention weights to be assigned to each instance predictions for each instrument label. The clip predictions are obtained using a weighted sum operation over the instance-level predictions using the outputs from the attention layer as weights.

weights are computed by normalizing the outputs of a fully connected layer, the weights of which correspond to parameters $\boldsymbol{v}$ in equation 3.4. Note that the output dimension of these two parallel fully connected layers is equal to the number of labels, i.e., 20. Finally, the attention weights are applied to the instance-level scores as shown in equation 3.3 to obtain the bag-level score.

Figure 3.3 illustrates the model architecture. The number of hidden units in each embedding layer is 128. Adding a skip connection from the input to the final embedding was found to stabilize the training across different random seeds. Batch normalization, ReLU activations, and a dropout of 0.6 is used after each embedding layer.

Training Procedure

The model performs a multi-label classification over 20 labels given an input. However, the OpenMIC dataset does not contain all labels for each instance. This leads to missing ground truth labels for training with loss functions such as binary cross-entropy (BCE). To account for this, I utilize the partial binary cross-entropy (BCE_p) loss function introduced for handling missing labels [105]:

$$\begin{aligned} \text{BCE}_p(\boldsymbol{y}, \boldsymbol{q}) &= \frac{g(p_y)}{L} \sum_{l \in L^o} \boldsymbol{y}_l \log \boldsymbol{q} + (1 - \boldsymbol{y}_l) \log(1 - \boldsymbol{q}) \\ g(p_y) &= \alpha p_y^{\gamma} + \beta \end{aligned} \tag{3.5}$$

Here $g(p_y)$ is a normalization function, p_y is the proportion of observed labels for the current data point, L is the total number of labels, L^o is the list of observed labels for the input data, $\boldsymbol{y}_l \in \{0, 1\}$ is the ground truth (absent or present) for label l, and $\boldsymbol{q}$ is the model's probability output for the label l being present in the input data $\mathbf{X}$. The hyperparameters in Eq. (3.5) are α, β, and γ. Note that in the absence of $g(p_y)$, data points with few observed labels will have a lower contribution in loss computation than those with several observed

labels. This is undesirable behavior and the inclusion of a normalization factor, dependent on the proportion of observed labels, is important. Here, α, β, and γ are set to 1, 0, and -1, respectively. This normalizes the loss for a data point by the number of observed labels and is equivalent to only computing the loss for observed labels.

Finally, the Adam optimization algorithm [106] is used for training with a batch size of 128 and learning rate of $5e^{-4}$ for 250 epochs. The model checkpoint is obtained at the epoch with the best validation loss. The training and testing splits released along with the OpenMIC dataset are used in order to foster reproducibility and comparability. The split was performed in a way such that clips from the same artists did not appear in the training and testing set [82]. Additionally, 15% data from the training split is randomly selected to create a validation set.

3.3 Experiments

3.3.1 Baselines

I use the following baseline systems for comparing the proposed attention-based model (ATT). Note that aside from the RF_BR model, all models are derived from state-of-the-art MIC model architectures.

1. RF_BR: This model is the baseline random forest model in [82]. A binary-relevance transformation is applied to convert the multi-label classification task into 20 independent binary classification tasks [107].
2. FC: A 3-layer fully connected network trained to predict the presence or absence of all instruments for a given data instance. Here, the input features of dimension 10×128 are flattened into a single feature vector for classification. Dropout is used for regularization and the Leaky ReLU (0.01 slope) is used. The model has 986772 parameters. Note that this model architecture is comparable to a deep CNN architecture by virtue of the VGGish model. Such CNN architectures have been established as state-of-the-art for predominant MIC [18] and strongly labeled MIC [20,

55].

3. FC_T: This model serves as an ablation study to observe the benefits of the attention mechanism. FC_T uses the same embedding layer as ATT. However, the aggregation of instance-level predictions is simply performed with average-pooling. The model has 52116 parameters. Similar to the FC model, this can be compared to a deep CNN architecture with global average pooling.
4. RNN: A 3-layer bi-directional GRU model with 64 hidden units per direction. The model processes the input features and produces a single 64-dimensional embedding. The embedding is passed to a 20-class classifier using a fully connected layer with a sigmoid activation function. The model has 226068 parameters. This model architecture is comparable to a deep CRNN model architecture which achieves state-of-the-art performance for other weakly labeled audio classification tasks such as music tagging [31], and sound event classification [89].

For each model, we train 10 randomly initialized instances with different random seeds and compute the classification metrics for each. This gives us a distribution of each model's performance. One benefit of ATT over the FC and RNN models is its small size. Both the ATT and FC_T utilize weight-sharing for embedding instances from the bags. This leads to significantly fewer learnable parameters.

3.3.2 Results and Discussion

Table 3.1 shows the overall performance of ATT compared to the baseline models using the overall precision, recall, F1-score, and AUROC. Additionally, we compare the instrument-wise F1-score for each model in Figure 3.4.

It can be observed that while the attention mechanism does not lead to an improvement in overall precision compared to the other models, the overall recall improved considerably and consequently the F1-score is higher. Note that while the difference between the metrics is small, the task is a 20-class classification problem with small differences still being

Table 3.1: Overall performance comparison between the proposed ATT model and four baseline models. All numbers are percentages.

	Overall Precision	Overall Recall	Overall F1-score	AUROC
RF_BR	**82.10** $\pm$ 0.20	77.69 $\pm$ 0.12	78.52 $\pm$ 0.14	77.69 $\pm$ 0.12
FC	80.62 $\pm$ 0.27	79.09 $\pm$ 0.20	79.19 $\pm$ 0.22	87.73 $\pm$ 0.15
FC_T	81.21 $\pm$ 0.11	79.79 $\pm$ 0.12	80.23 $\pm$ 0.11	88.87 $\pm$ 0.08
RNN	81.18 $\pm$ 0.30	79.55 $\pm$ 0.31	79.93 $\pm$ 0.32	88.42 $\pm$ 0.13
ATT	81.65 $\pm$ 0.18	**80.68** $\pm$ 0.17	**80.84** $\pm$ 0.18	**89.08** $\pm$ 0.08

statistically significant. Here the difference between the ATT model performance and other models is statistically significant. We observe that ATT performs better than RF_BR in almost every instrument label, especially for the labels with high positive-negative class imbalance, such as clarinet, flute, and organ. This ties to the observation made about improved recall, as ATT is able to overcome this imbalance possibly due to the ability to localize the relevant instances for the minority class. In the case of an imbalanced instrument label, the recall for the minority class greatly suffers for RF_BR. While this problem is easily mitigated in standard multi-class problems by using balanced sampling, it is not straightforward to address with multi-label data. Comparing to FC_T, we can attribute the improved performance of ATT to better aggregation of instance-level predictions. FC_T is essentially the same model as ATT using mean pooling instead of attention, and ATT outperforms it for most instrument classes, especially the more difficult to classify instruments. The RNN model also outperforms the RF_BR baseline. In polyphonic music, the instances in a bag are structured and highly correlated and hence using a recurrent network to model the temporal structure in the instance sequence leads to a powerful embedding of the bag, incorporating useful information from each instance.

To further study the effect of class imbalance on model performance, an experiment using balanced cross-entropy loss is performed. The standard binary cross-entropy loss function for a single label can be written as follows:

$$\text{BCE}(\boldsymbol{y}, \boldsymbol{q}) = \boldsymbol{y} \log \boldsymbol{q} + (1 - \boldsymbol{y}) \log(1 - \boldsymbol{q}), \tag{3.6}$$

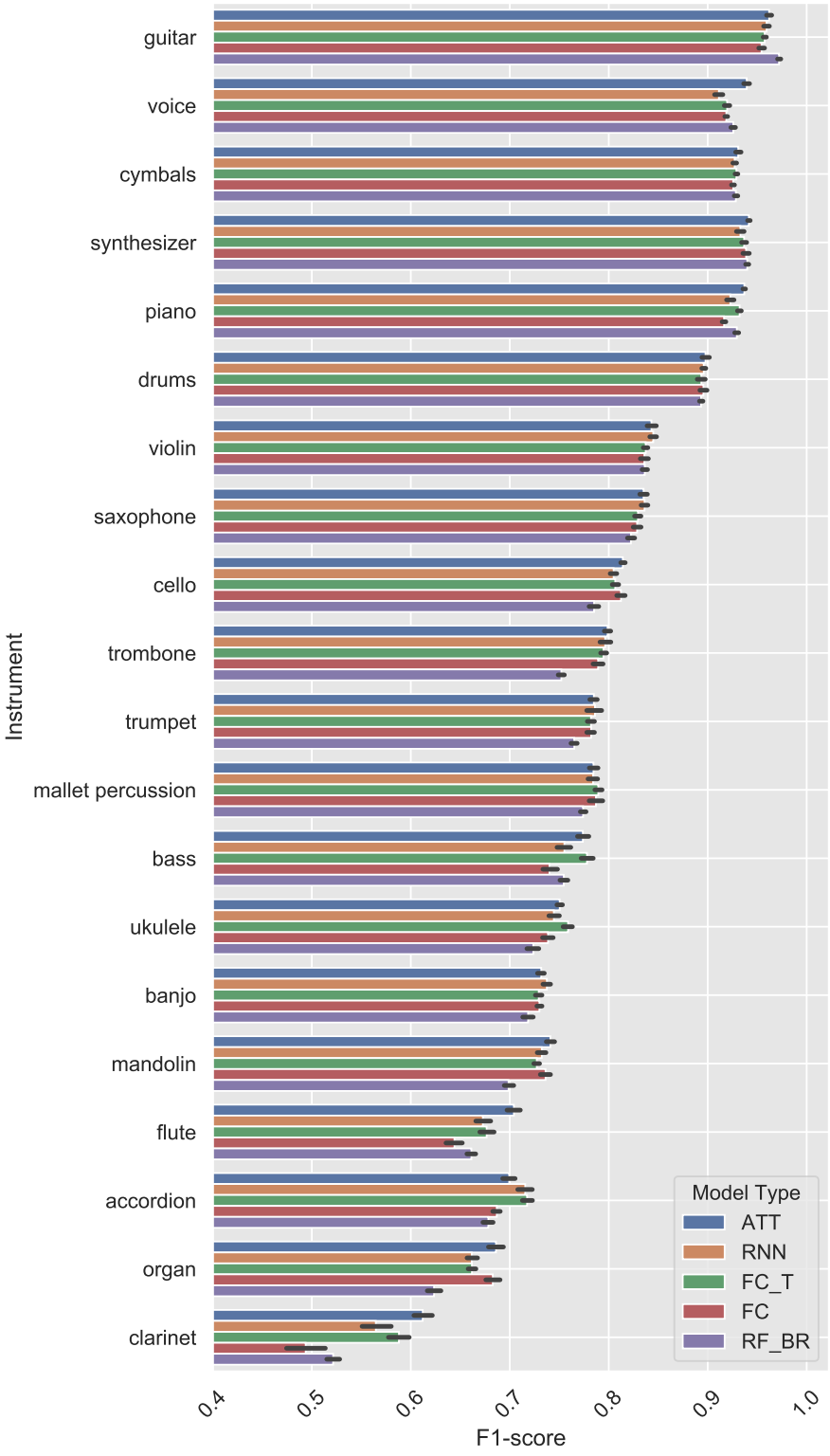

Figure 3.4: Instrument-wise Macro-averaged F1-scores. ATT is the proposed model. RNN is a 3-layer GRU network. FC_T is similar to ATT except that it utilizes mean pooling instead of attention. FC is a fully connected network. RF_BR is a collection of 20 random forest binary classifiers.

where $\boldsymbol{y}$ is the ground truth label, and $\boldsymbol{q}$ is the prediction. In the balanced binary cross-entropy loss function, a weight is applied to balance the contribution of positive and negative classes. It can be written as follows:

$$\text{BCE}_{bal}(\boldsymbol{y}, \boldsymbol{q}) = w_p \boldsymbol{y} \log \boldsymbol{q} + (1 - \boldsymbol{y}) \log(1 - \boldsymbol{q}), \tag{3.7}$$

$$w_p = \frac{\text{Neg}}{\text{Pos}} \tag{3.8}$$

where w_p is the weight applied to the positive label and is calculate as the ratio of negative instances to positive instances. Note that the partial binary cross-entropy loss is weighted in this case. The experiment involves various settings of utilizing the balanced loss function. First, the balanced loss function is used from the start of training, with the same hyperparameters as those used for training the ATT model. Second, the ATT model trained previously is fine-tuned for 50 additional epochs using the balanced loss. This ensures that the retraining process starts with a good initial model for MIC. Finally, only the attention head and prediction head of the previously trained ATT model is fine-tuned. This implies that the embeddings layers remain fixed during training. This process is commonly used in transfer learning to ensure that the model retains the intermediate representations and only updates the final layers to account for imbalance in the data. Table 3.2 shows the results of this experiment. Training with the balanced loss is found to be detrimental to the overall performance of the model, especially when training from scratch. This experiment, however, shows that precision and recall can indeed be tuned to a certain extent using the balanced loss term. Additionally, more complicated class imbalanced learning methods may need to be utilized to improve overall performance. Note that, this experiment is not conducted for the other proposed methods as no improvement is observed.

We visualize the attention weights for two example clips in Figure 3.5. The left clip is from the test set and starts with the vocals fading out until 2 seconds. From 5s onwards, the vocals grow in loudness until the end of the clip. The violin plays throughout but is the

Table 3.2: Model performance under different settings of balanced loss during training. Balanced loss is used to address positive-negative imbalance in different instruments. 'BL' refers to balanced loss, where the binary cross-entropy loss function is weighted according to the ratio of positive to negative instances. 'FT' refers to fine-tuning, where the model is initially trained using an unweighted loss and fine-tuned later with the balanced loss function. 'FT final' refers to fine-tuning only the attention and prediction heads of the ATT architecture. This ensures that the embedding layers remain unchanged.

	Overall Precision	Overall Recall	Overall F1-score	AUROC
ATT	81.65 ± 0.18	80.68 ± 0.17	80.84 ± 0.18	89.08 ± 0.08
BL	81.87 ± 0.19	76.40 ± 0.23	77.05 ± 0.30	88.81 ± 0.16
BL FT	80.40 ± 0.08	81.50 ± 0.14	79.40 ± 0.29	89.04 ± 0.06
BL FT final	80.95 ± 0.29	81.34 ± 0.07	80.51 ± 0.32	89.03 ± 0.02

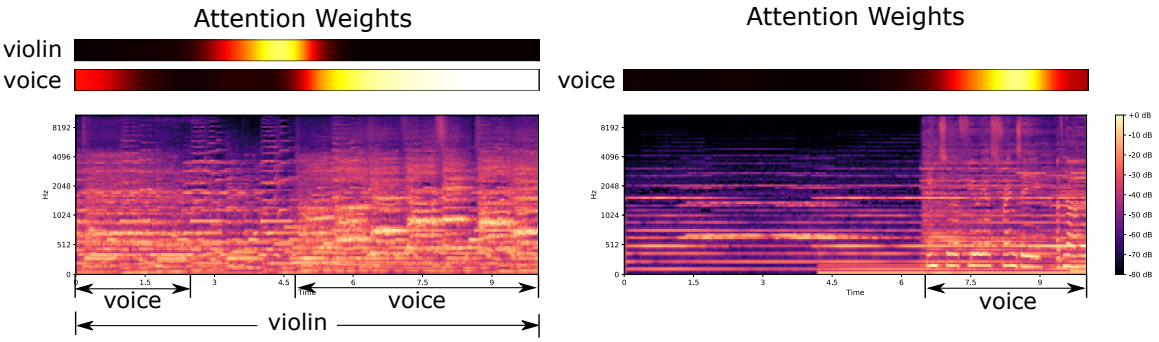

Figure 3.5: Attention Weight Visualization: The horizontal bars above the mel-spectrogram represent the attention weights across the instances of the clip for the respective instruments.

predominant instrument only for a few seconds between 3 and 6 seconds, as visualized in the corresponding attention weights as well. The right clip is from the training set and contains vocals starting from 6s onwards. The attention weights for vocals directly coincides with that. An interesting observation is that the annotation for vocals was missing for this clip.

3.4 Conclusion

This chapter focused on addressing the challenge of weakly labeled data for the MIC task. I framed the MIC task as a MIML problem, which relies on aggregation of instance-level predictions to predict the final clip-level prediction. A simple attention-based aggregation mechanism is utilized which shows improved performance against several baseline models.

In general, formulating the weakly labeled MIC task in the MIML framework and

utilizing smart aggregation strategies outperforms simpler modeling strategies that treat the entire input as a single feature. This trend is not only observed for the ATT model but also for the FC_T model which uses a much simpler aggregation strategy — average pooling — compared to attention.

All datasets for MIC except the OpenMIC dataset have strong labels. The prevalence of SLDs influenced algorithms for instrument recognition leading to primarily strongly supervised methods which are incapable of being trained effectively using weakly labeled data — although trained MIC systems have indeed been evaluated with weakly labeled data (see IRMAS dataset in Section 2.2). The success of the proposed method in leveraging the weakly labeled data is an exciting prospect since it encourages the collection of more weakly labeled data for MIC, possible leading to even larger datasets for MIC. Additionally, the MIL framework is applicable in other MIR Tasks, especially weakly labeled music classification problems, such as music tagging or emotion classification. The MIL framework may also potentially be applicable in interpretable and explainable machine learning systems in MIR, which is an increasingly popular topic [108, 109, 110], as demonstrated by the Figure 3.5.

One drawback of this method is the treatment of all instances and labels as independent. Temporal dependence may be ignored by the instance independence assumption in the MIL framework. As for label correlation, since the models are trained in a multi-label manner, with weight sharing in the embedding layers there is at least some degree of shared representation learning for different instrument labels. However, any label correlations will be learned implicitly in this case. Explicitly modeling label-correlation in multi-label classification has shown to significantly improve the classification performance [111, 112, 113, 113]. However, exploring ways to incorporate label-correlation for MIC with the OpenMIC dataset has the additional challenge of missing and sparse labels [114] and is therefore subject of future work. The label independence problem is not restricted to instrument classification. Most multi-label audio classification tasks use a very similar setup and do not usually try to explicitly model label correlations. Finally, the MIC task

can be formulated as a MIML problem **with missing labels**. The ATT model ignores any information that may be obtained from the missing instrument labels.

CHAPTER 4

GENERATIVE MODELING FOR SEMI-SUPERVISED LEARNING

A vast majority of research in MIC utilizes discriminative models with fully labeled data. Generative models have the benefit of not required all data to be labeled and can therefore provide ways to leverage the OpenMIC dataset which is a partially labeled dataset. This chapter focuses primarily on SSL using a generative modeling approach, aiming to answer both RQ2 and RQ3.

4.1 Semi-Supervised MIC

Having discussed the attention-based model to address the challenge of weakly labeled data in MIC, I move on to the other challenge listed in Section 2.2.2: Missing or unlabeled data in the OpenMIC dataset. Missing labels are not a new problem in ML. Domains such as computer vision (CV) and natural language processing (NLP), where it is impractical to label all objects/topics in all images/documents in a dataset face a similar problem. Over the years, several methods have been proposed to address the problem of limited labeled data in the ML community [115, 114, 62, 116]. These methods fall under the umbrella of semi-supervised learning (SSL).

Missing labels are especially problematic in the case of MIC because of the following two issues with the OpenMIC dataset:

1. Smaller size of the dataset: The dataset consists of 20,000 clips as compared to hundreds of thousands of images in CV datasets such as MS-COCO (330,000 images).

2. Small proportion of labeled data: The OpenMIC dataset has 20 instrument classes and 20,000 audio clips. Had the dataset been fully labeled there would be 400,000 (20×20000) labels available. However, the total number of labels available is only

about 40,000. This means around that 10% of the data is labeled.

All methods discussed thus far (see sections 3 and 2.4) can only utilize labeled data since they are all supervised learning models trained using loss functions such as binary cross-entropy which requires labels. There may, however, be information that models may be able to utilize from their predictions for unlabeled data as well. In the most general setting of SSL, small amounts of labeled data and large amounts of unlabeled data are utilized in tandem during training of models. The task of MIC using the OpenMIC dataset lends itself well to the SSL framework since it involves both labeled and unlabeled data.

4.1.1 Background

In addition to MIC, several application domains such as image search [117], genomics [118], and speech analysis [119], face the challenge that only a small subset of the observed data is labeled. The process of labeling the remaining data is usually very expensive and therefore not considered an option. The small amount of labeled data makes it problematic to simply employ supervised learning methods due to issues such as overfitting to the labeled subset leading to poor generalizability. SSL algorithms leverage both, the labeled subset of the data for supervised learning, as well as the unlabeled subset of the data with the goal to improve classification performance over purely supervised models.

In the *self-training* method [120], highly confident model predictions for the unlabeled data are used to bootstrap the training procedure until a termination criterion is reached. *Co-training* is an extension of self-training where multiple models utilize different feature representations of the data to fit the labeled training data [121, 122]. These methods face the issue of reinforcing poor predictions. *Graph-based* methods are among the most popular methods for SSL. In this paradigm, graphs are constructed consisting of nodes for labeled and unlabeled data. The nodes are connected by edges representing similarity between them. Label information flows from labeled to unlabeled nodes based on minimizing energy using maximum a posteriori (MAP) estimation [123, 124].

More recently, neural network-based SSL has grown in popularity. A standard approach consists of supervised neural network classifiers with additional penalty terms from an unsupervised autoencoder component [125]. Kingma et al. devised deep generative models for semi-supervised learning based on Variational Autoencoders (VAEs) [60] which is scalable to large datasets [61]. Maaløe et al. extend the previously proposed deep generative models by introducing an additional latent variable and skip connections in a 2-layer hierarchical deep generative model [126]. These are the auxiliary deep generative model and skip deep generative models, respectively.

Although data-driven research in MIR has largely focused on supervised learning methods, a few researchers utilized SSL methods for various MIR tasks. For single note MIC, Diment et al. utilize a Gaussian Mixture Model (GMM) trained with an iterative Expectation Maximization (EM)-based algorithm that incorporates both labeled and unlabeled data to estimate the GMM parameters [127].They found that the semi-supervised GMM outperformed purely supervised GMM. The task of singing voice detection has been addressed using SSL-based methods such as co-training [128], and student-teacher learning [129]. Wu et al. utilized student-teacher learning for automatic drum transcription [130]. Self-training was used for semi-supervised onset-detection, improving performance over hand-tuned onset detectors [131]. While deep generative models are often utilized in music and audio generation [132, 133], there has not been any work investigating generative modeling as a method for SSL-based music classification.

4.2 Problem Formulation

Again, the MIC task involves predicting the presence or absence of pre-determined instruments in input audio. Formally, this is a multi-label classification problem. Additionally, the training data (OpenMIC dataset), is not fully labeled, i.e., presence or absence of only a few instruments is known for a given data point. This makes the MIC task a multi-label classification with missing labels problem.

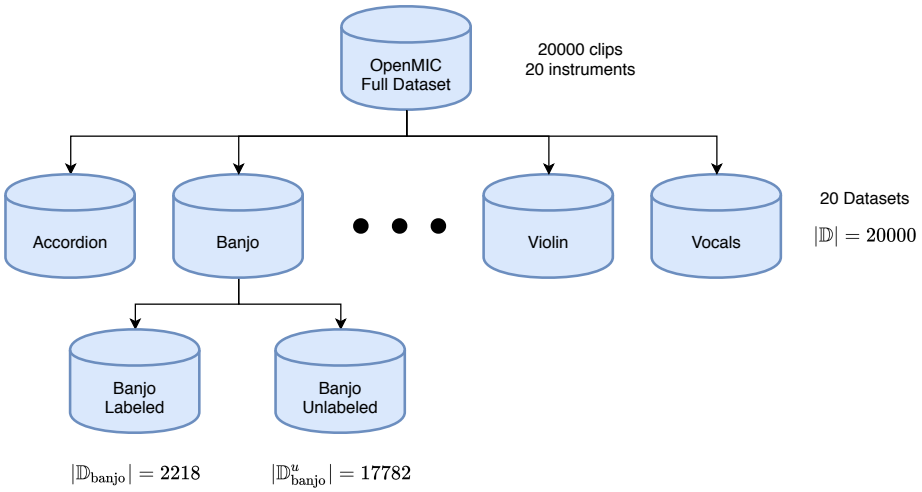

Figure 4.1: Binary relevance decomposition: The multi-label OpenMIC dataset containing 20 instrument labels can be regarded as 20 binary classification datasets, one for each instrument label. Transforming the multi-label problem into binary classification problems is called binary relevance decomposition. In the OpenMIC case, since not all data points have positive or negative class annotations for each instrument, this means that each of the binary classification datasets can be partitioned into a labeled and unlabeled subset. This setup lends itself well for the SSL framework.

To utilize deep generative models for SSL, I simplify the multi-label classification problem into multiple binary classification problems; one for each instrument. This is known as binary relevance decomposition [134, 107]. This is similar to the way the RF_BR baseline model is trained in Section 3.3.1. A dataset for multi-label classification may be represented as $\mathbb{D} = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots (\boldsymbol{x}_m, \boldsymbol{y}_m)\}$ where $\boldsymbol{x}_i$ is the feature vector representing a data point and $\boldsymbol{y}_i \in \{-1, 0, 1\}^L$ is the label vector for the data point. $\boldsymbol{y}_{i,j} \in -1, 1$ when the label j is relevant to $\boldsymbol{x}$. The value of $\boldsymbol{y}_{i,j}$ indicates whether the label is positively or negatively associated. $\boldsymbol{y}_{i,j} = 0$ implies that label j is irrelevant to the $\boldsymbol{x}$ or missing. In the binary relevance decomposition, the multi-label dataset $\mathbb{D}$ can be divided into several binary subsets $\mathbb{D}_l$ such that $\mathbb{D} = \bigcup_{i=1}^{L} \mathbb{D}_l$ where $\mathbb{D}_l = \{(\boldsymbol{x}_1, \boldsymbol{y}_{1,l}), \ldots (\boldsymbol{x}_n, \boldsymbol{y}_{n,l})\} \ni \boldsymbol{y}_{i,l} \neq 0$. In the context of MIC, the OpenMIC dataset is divided into 20 different datasets with each dataset consisting only of audio clips where the corresponding instrument is labeled as present or absent.

Note that for a given instrument I, the complete OpenMIC dataset can be partitioned into

$\mathbb{D} = \mathbb{D}_I \dot{\cup} \mathbb{D}_I^u$ where $\mathbb{D}_I^u$ consists of audio clips from the OpenMIC dataset that do not have a label for instrument I, i.e. they are unlabeled. Thus, for each instrument, the OpenMIC dataset has a labeled and unlabeled subset which can be used to train a binary classifier using SSL. Figure 4.1 illustrates the binary relevance decomposition process for the OpenMIC dataset. The remainder of this chapter focuses on semi-supervised binary classification for a single instrument. Thus, the instrument index I is dropped for clarity. The labeled subset is referred to as $\mathbb{D}_l$ and the unlabeled subset is referred to as $\mathbb{D}_u$. $\mathbb{D}_l$ and $\mathbb{D}_u$ are used in the SSL framework to study the usefulness of generative modeling-based SSL for the MIC task.

4.3 Semi-Supervised Deep Generative Model

Kingma et al. introduced a framework for generative model-based semi-supervised learning which utilizes neural networks as rich density estimators [61]. The model they propose is a probabilistic latent variable model[1] trained using stochastic variational inference which demonstrates the ability to separate data classes from variabilities that may be present within the same class. I utilize this framework for my experiments with generative SSL models since the use of stochastic variational inference allows models to be trained with large amounts of data. This section provides the mathematical background for the semi-supervised deep generative model (DGM), while tying it to the MIC task.

The proposed model describes the data $\mathbf{x}$ as being generated by a class variable y and a latent variable $\mathbf{z}$. The following generative process is used:

$$\begin{aligned} p(y) &= \text{Cat}(y|\boldsymbol{\pi}) \\ p(\mathbf{z}) &= \mathcal{N}(\mathbf{z}|\mathbf{0}, \boldsymbol{I}) \\ p_\theta(\mathbf{x}|y, \mathbf{z}) &= f(\mathbf{x}; y, \mathbf{z}, \boldsymbol{\theta}), \end{aligned} \tag{4.1}$$

where $\text{Cat}(y|\boldsymbol{\pi})$ is a multinomial distribution in general but binomial in the case of MIC since instruments are either present or absent. In the case of unlabeled data, y is also treated as a

[1] Kingma et al. refer to this model as M2 [61]

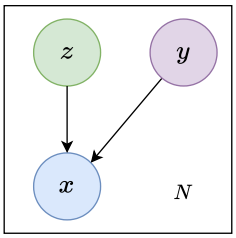

Figure 4.2: Plate notation for the semi-supervised DGM. $\mathbf{x}$ is modeled as a function of latent $\mathbf{z}$ and class label y. When y is missing, or the data is unlabeled, then y is also treated as a latent variable.

latent variable. y and $\mathbf{z}$ are marginally independent variables enabling the generative model to separate the class identity from latent factors such as genre or tempo in the case of MIC, or handwriting style in the case of digit classification. Figure 4.2 illustrates the graphical model for the generative process. $f(\mathbf{x}; y, \mathbf{z}, \boldsymbol{\theta})$ is the likelihood function parameterized by a deep neural network. During the inference process, an integral is performed over the classes for unlabeled or unobserved y. While the integration step may render this method impractical for problems with large number of classes, the decomposed MIC problem is a simple binary (2-class) problem. The missing labels are predicted using the inferred posterior distribution $p_\theta(y|\mathbf{x})$.

4.3.1 Evidence Lower Bound Objective (ELBO)

The true posterior distribution:

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p(\mathbf{x})} = \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{\int_z p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}} \tag{4.2}$$

is intractable to compute (without making any simplifying assumptions). The reason being that, unlike the categorical variable y, $\mathbf{z}$ is a continuous variable and the inference process requires integrating over all possible configurations of $\mathbf{z}$. VAEs were introduced to solve this problem by approximating the true posterior with a fixed-form posterior distribution $q_\phi(\mathbf{z}|\mathbf{x})$ parameterized by $\boldsymbol{\phi}$. To ensure that the approximate posterior $q_\phi(\cdot)$ is as close to the true posterior $p(\mathbf{z}|\mathbf{x})$ a lower bound on the marginal likelihood of the model is derived

following the variational principle. This lower bound is used as the objective function to be optimized during training. In VAEs, $q_\phi(\cdot)$ serves as an inference model or *encoder* to estimate a distribution (usually Gaussian) for the latent code $\mathbf{z}$ given data $\mathbf{x}$. The main benefit of VAEs is that the encoder network estimates a shared set of global variational parameters $\boldsymbol{\phi}$ for all data points instead of computing variational parameters for each data point. This is known as amortized inference and it allows fast inference for training and testing.

For the deep generative semi-supervised model, an encoder is introduced for the latent variables y, and $\mathbf{z}$. These encoders are specified as Gaussian and Multinomial distribution respectively.

$$q_\phi(\mathbf{z}|y,\mathbf{x}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_\phi(y,\mathbf{x}), \mathrm{diag}(\boldsymbol{\sigma}^2_\phi(\mathbf{x}))), \tag{4.3}$$

$$q_\phi(y|\mathbf{x}) = \mathrm{Cat}(y|\boldsymbol{\pi}_\phi(\mathbf{x})), \tag{4.4}$$

Here, $\boldsymbol{\mu}_\phi(\cdot)$ is a vector of Gaussian means, $\boldsymbol{\sigma}_\phi(\cdot)$ is a vector of standard deviations, $\boldsymbol{\pi}_\phi(\cdot)$ is a probability vector, and the functions $\boldsymbol{\mu}_\phi(\cdot)$, $\boldsymbol{\sigma}_\phi(\cdot)$ and $\boldsymbol{\pi}_\phi(\cdot)$ are implemented as fully connected DNNs.

While constructing the objective function, there are two cases that need to be taken into consideration. The lower bound on the marginal distribution for labeled data $\mathbb{D}_l$ and for unlabeled data $\mathbb{D}_u$. For $\mathbb{D}_l$, the lower bound can be written as:

$$\begin{aligned}\log p_\theta(\mathbf{x},y) &\geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},y)}\left[\log p_\theta(\mathbf{x}|y,\mathbf{z}) + \log p_\theta(y) + \log p(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x},y)\right] = -\mathcal{L}(\mathbf{x},y) \\ -\mathcal{L}(\mathbf{x},y) &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},y)}\left[\log p_\theta(\mathbf{x}|y,\mathbf{z}) + \log p_\theta(y)\right] - \mathrm{KL}\left[q_\phi(\mathbf{z}|\mathbf{x},y)||p(\mathbf{z})\right] \end{aligned} \tag{4.5}$$

Here, the model for $\log p_\theta(\mathbf{x}|y,\mathbf{z})$ is referred to as the *decoder* which reconstructs the input data.

As mentioned earlier, in the case of unlabeled data, y is also treated as a latent variable along with z. In this case, posterior inference is carried out over all values of y and the

following lower bound is obtained:

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{q_\phi(y,\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|y,\mathbf{z}) + \log p_\theta(y) + \log p(\mathbf{z}) - \log q_\phi(y,\mathbf{z}|\mathbf{x})\right] = -\mathcal{U}(\mathbf{x})$$

$$-\mathcal{U}(\mathbf{x}) = \sum\nolimits_y q_\phi(y|\mathbf{x})(-\mathcal{L}(\mathbf{x},y)) + \mathcal{H}(q_\phi(y|\mathbf{x})) \tag{4.6}$$

Thus, the loss function for the entire dataset may be written as follows:

$$\mathcal{J} = \sum\nolimits_{(\mathbf{x},y)\sim\mathbb{D}_l} \mathcal{L}(\mathbf{x},y) + \sum\nolimits_{\mathbf{x}\sim\mathbb{D}_u} \mathcal{U}(\mathbf{x}) \tag{4.7}$$

4.3.2 Auxiliary Classification Loss

Note that the distribution $q_\phi(y|\mathbf{x})$ in equation 4.4 can also be interpreted as a discriminative classification network. This model is used to infer the missing label for unlabeled data, and also as the classifier during test time. Since DGM is being applied to a classification problem, the goal of this method is to obtain the best possible classifier $q_\phi(y|\mathbf{x})$.

However, in equation 4.7, $q_\phi(y|\mathbf{x})$ is only being optimized for the second term, i.e., for the unlabeled data $\mathbb{D}_u$. This is far from ideal since the classifier can be improved using the available labeled data $\mathbb{D}_l$. Kingma et al. propose the use of an auxiliary classification loss in the objective function. Thus $q_\phi(y|\mathbf{x})$ can learn from the entire dataset, and all model and variational parameters are trained for each data point regardless of whether it is labeled or not. The final loss function is as follows:

$$\mathcal{J}^\alpha = \mathcal{J} + \alpha \cdot \mathbb{E}_{\mathbb{D}_l}\left[-\log q_\phi(y|\mathbf{x})\right], \tag{4.8}$$

where α is a hyperparameter that essentially determines the contribution of the purely discriminative loss.

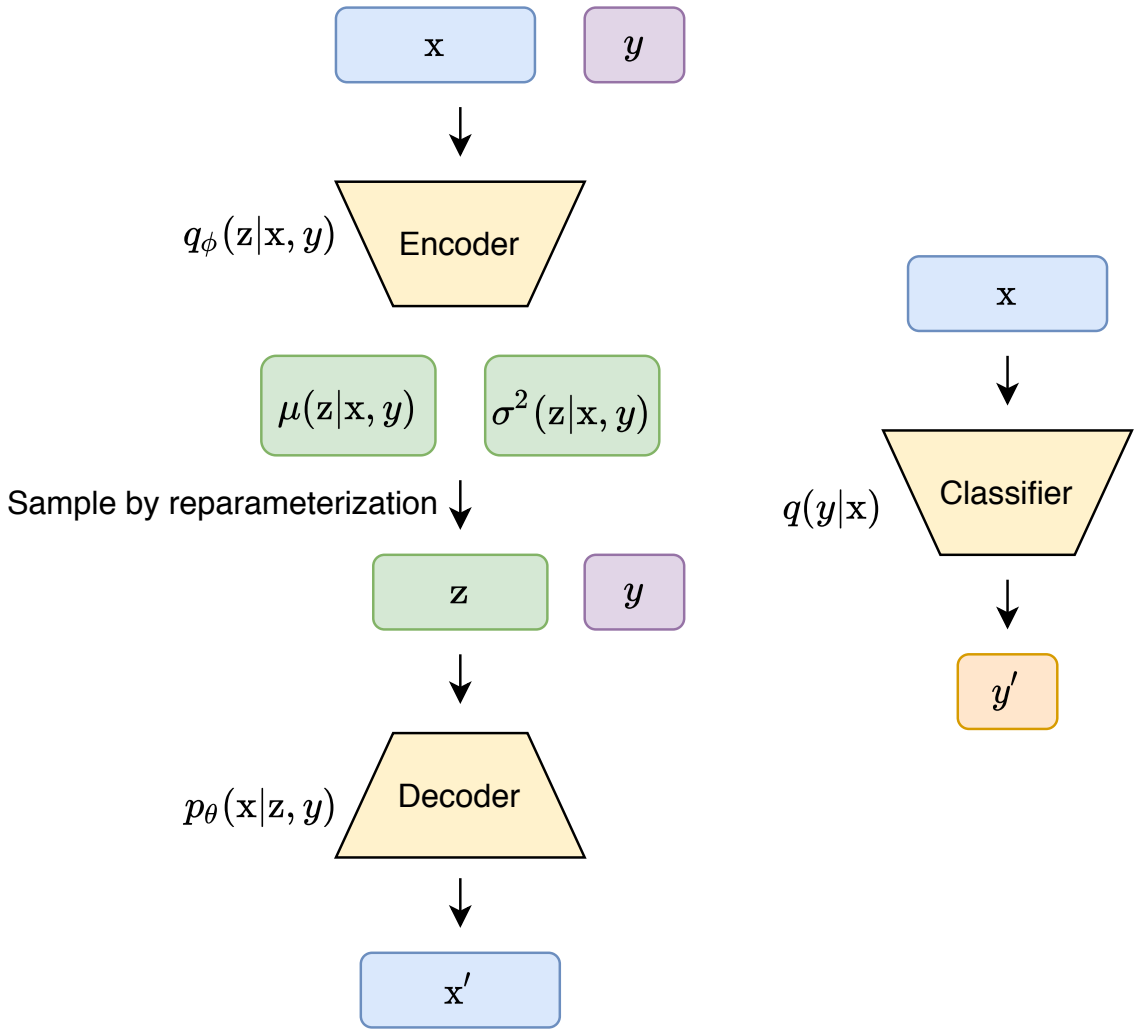

Figure 4.3: DGM model architecture showing the semi-supervised VAE and classification subnetworks

DGM model architecture. On the left is the semi-supervised VAE and on the right is the classifier which is trained in parallel on both labeled and unlabeled data. Blue blocks represent vectors in the input feature space x, x'. Green blocks represent latent variables, and their distributions $z, \mu(z|x, y), \sigma^2(z|x, y)$. Purple blocks represent partially observed class y. For labeled data, y is the observed class categorical variable (one-hot). For unobserved or unlabeled data, posterior inference is performed over all possible ys (instrument present or absent). The orange y represents the predicted class label which is used to calculate $q_\phi(y|\mathbf{x})$. All trapezoids in yellow represent fully connect networks.

Table 4.1: Encoder and Decoder hyperparameters for the DGM. The h1, h2, and h3 refer to the choice of the hidden layer sizes. Each layer is followed by a ReLU non-linearity.

Encoder and Decoder
h1 = $\{256, 512\}$
h2 = 256
h3 = $\{0, 256\}$

4.4 Model Architecture

The previous section covers the theory behind the semi-supervised DGM. Figure 4.3 illustrates the model architecture. I use the same VGGish features as described in Section 3.2 as input $\mathbf{x}$ to the model. The input to the encoder is the input $\mathbf{x}$ concatenated with a one-hot encoding of the class y. For unlabeled data, the positive and negative class one-hot vectors are concatenated to $\mathbf{x}$ to form two separate inputs to the encoder for each unlabeled data point.

The encoder estimates the variational parameters: Gaussian mean $\boldsymbol{\mu}$ and log-variance $\log \boldsymbol{\sigma}^2$. The re-parameterization trick [60] is applied to sample a latent vector $\mathbf{z}$. The decoder input is a concatenation of $\mathbf{z}$ and one-hot encoding of y for the labeled data. Similar to the encoder, for unlabeled data both positive and negative class encodings are concatenated to $\mathbf{z}$. The decoder finally outputs the reconstructed data $\mathbf{x}'$.

As mentioned in Section 4.2, the multi-label MIC task is transformed into several (20 in the OpenMIC case) binary instrument classification tasks. 20 different binary classifiers are trained, one for each of the instrument classes in the OpenMIC dataset. For the model architecture, I perform a grid search on the depth and width of the encoder and decoder hidden layers. The choice of hyperparameters is shown in Table 4.1.

The best model architecture for each instrument model is chosen based on performance on validation data, across 5 random initializations. This implies that the model architecture for each instrument may differ. I fix the latent $\mathbf{z}$ dimension to 64. The classifier $q_\phi(y|\mathbf{x})$ architecture is a simple feed-forward network with 3 hidden layers with dropout as a regularization method. This architecture is similar to that used in the FC model in

Section 3.3.1.

4.4.1 Training Procedure

The ELBO is optimized using the Adam optimization algorithm [106]. The learning rate is chosen between $0.00009, 0.0001, 0.0002$. β_1 and β_2 are set to 0.9 and 0.999, respectively. A weight decay of 1×10^{-5} is used for regularization. A mini-batch size of 128 is chosen for optimization. Grid search is performed to obtain the classification loss weight α. The classification loss weight is chosen as $\alpha = \beta \cdot \frac{|\mathbb{D}_u|}{|\mathbb{D}_l|}$ where $\beta \in 16, 32, 64$. Each model is trained for 100 epochs. In this experiment, an epoch is completed after one full pass of $\mathbb{D}_u$ is completed. Since $|\mathbb{D}_u| > |\mathbb{D}_l|$, it is possible that multiple full passes of $\mathbb{D}_l$ occur during one epoch. Algorithm 1 explains the training loop. As for the model architecture, the best hyperparameters for each instrument classifier are selected based on performance on validation data across 5 runs.

Algorithm 1: Training Loop for DGM: N is the number of epochs, $\boldsymbol{X}_u$ is a minibatch of unlabeled data, and $(\boldsymbol{X}, \boldsymbol{Y})$ is a minibatch of labeled data.

Result: Optimized DGM parameters $\hat{\theta}, \hat{\phi}$
Input: $\mathbb{D}_l, \mathbb{D}_u$ Initialize θ, ϕ;
for $i = 1 \dots N$ **do**
  forall $\boldsymbol{X}_u \in \mathbb{D}_u$ **do**
    sample $(\boldsymbol{X}, \boldsymbol{Y})$ from $\mathbb{D}_l$;
    loss $= -ELBO((\boldsymbol{X}, \boldsymbol{Y}), \boldsymbol{X}_u)$;
    Gradient descent to optimize θ, ϕ;
  end
end

4.5 Experiment

The aim of my experiment with this semi-supervised DGM is to study the effectiveness of generative modeling as a means of leveraging both labeled and unlabeled data for MIC, and thus deal with the missing label issue in the OpenMIC dataset. To that end I compare the semi-supervised DGM against two baseline models.

Table 4.2: Overall results for DGM versus supervised-only ATT and FC models. The mean and standard deviation of the results are shown across 5 seeds. All numbers are percentages.

	AUROC	Overall Recall	Overall F1-score	AUROC
ATT	**81.65** $\pm$ 0.18	80.68 $\pm$ 0.17	**80.84** $\pm$ 0.18	**89.08** $\pm$ 0.08
DNN	80.62 $\pm$ 0.27	79.09 $\pm$ 0.20	79.19 $\pm$ 0.22	87.73 $\pm$ 0.15
DGM	79.79 $\pm$ 0.16	**81.14** $\pm$ 0.11	79.54 $\pm$ 0.19	88.78 $\pm$ 0.05

4.5.1 Baselines

Both the baseline models I compare with are supervised discriminative models. Training only the classifier $q_\phi(y|\mathbf{x})$ using the same model architecture as described in Section 4.4 serves as a fair baseline where the only difference between the two models is the additional semi-supervised VAE. This, however, is treated as a weak baseline since the discriminative classifier has a simple architecture. In contrast, the attention-based model described in Section 3 is chosen as a strong baseline.

4.5.2 Evaluation

To properly evaluate the semi-supervised binary classifiers, the exact same training and test split is used as in previously described experiments. The binary relevance decomposition (see Section 4.2) is performed only on the training set. For validating each instrument's classifier, separate validation sets are created by randomly sampling 15% of the labeled set $\mathbb{D}_l$.

4.5.3 Results and Discussion

Table 4.2 and Figure 4.4 shows the performance of the DGM as compared to the two baseline models. Overall, the DGM outperforms the weak baseline but fails to outperform the strong baseline. Comparing the results instrument-wise, the DGM outperforms the weak baseline for 12 out of 20 instruments, but only manages to outperform ATT for 4 out of 20 instruments. This does not reflect well on the performance of the DGM as both the baselines utilize only the labeled subset of the OpenMIC dataset.

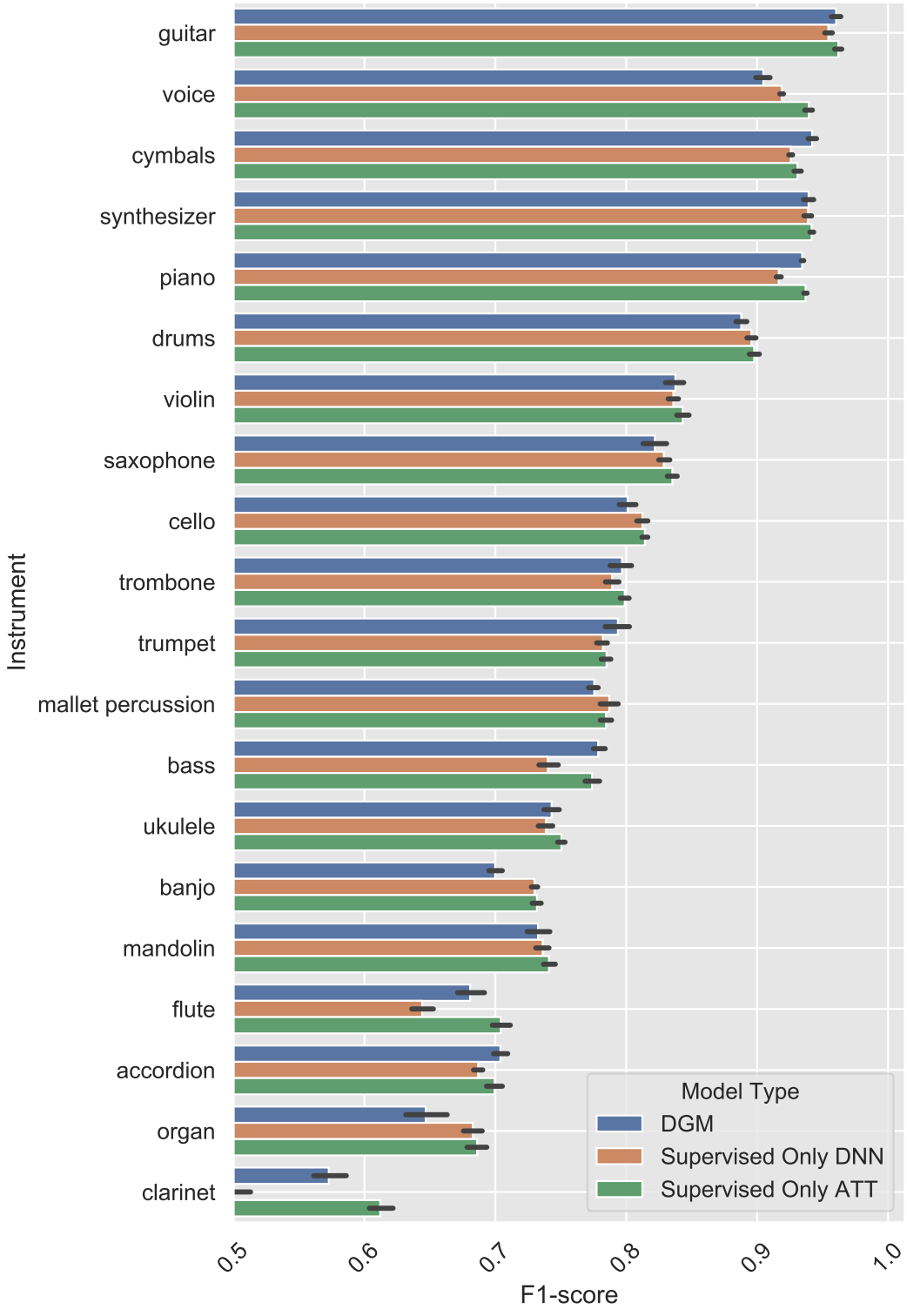

Figure 4.4: Instrument-wise Macro-averaged F1-scores comparing DGM against a simple DNN classifier which uses the same architecture as $q_\phi(y|\mathbf{x})$. The DGM is also compared against the ATT model from Chapter 3 which serves as a stronger supervised-only baseline.

There are a few explanations for why the DGM fails to outperform the ATT model, which only relies on the labeled subset of the dataset. First, the ATT model has the following advantages over the DGM:

(i) training as a multi-label classifier enables the ATT model to implicitly learn shared representations that benefit the classification of individual instruments, a concept similar to multi-task learning, and

(ii) the attention mechanism incorporates an inductive bias that proves to be useful for learning with weakly labeled data (see 3.3.2).

Second, Kingma et al. evaluate the DGM on larger and simpler image datasets such as MNIST and SVHN [61]. The OpenMIC dataset is not only smaller in size, but also a diverse dataset of audio. Generative models for music in the audio domain are very difficult to train, especially for diverse music with multiple instruments. Most music generation models are trained using datasets of single note or short phrases of single instrument music [132, 133]. In this experiment, a transfer learned feature space is used as the input representation of audio instead of lower-level audio representations such as raw audio or spectrograms. Since the goal of this experiment is to learn the best classifier trained using both labeled and unlabeled data adopting input features with strong discriminative power may be justified. However, this choice may be detrimental to the generative performance of the DGM. The benefits of DGM have mainly been demonstrated for simpler image domains which may be the reason why it does not succeed in modeling more complex music data. Additionally, the semi-supervised image classification evaluation strategy involves very few labels as compared to the proportion of labels in the OpenMIC dataset.

4.6 Conclusion

This chapter presented the first SSL-based approach to address the challenge of missing labels in the OpenMIC dataset. Most formulations of the MIC task utilize a fully labeled

training dataset. The ATT method presented in Chapter 3 is indeed trained with the partially labeled OpenMIC dataset but only the available labels are used during the actual model learning process since the ATT model is a supervised discriminative model. The DGM presented in this chapter is capable of learning using both the labeled and unlabeled portion of the OpenMIC dataset. First, the multi-label MIC task is decomposed into 20 different binary classification tasks, each with a corresponding labeled and unlabeled dataset. These datasets are used to train the semi-supervised DGM.

The model is a semi-supervised VAE which describes the generative process of the data $\mathbf{x}$ as a non-linear transformation of latent factors $\mathbf{z}$ sampled from a Gaussian, and class variable y. This works for both the labeled data directly. For unlabeled data, y is also treated as a latent variable and is inferred from a classifier which is trained in parallel. The semi-supervised model outperformed the simpler supervised classifier but falls short of the stronger baseline model. The lack in performance may be attributed to the fact that the generative model is not equipped with mechanisms to handle weakly labeled data, and that separate binary classifiers for each instrument tend to perform worse than a multi-label classifier.

CHAPTER 5

CONSISTENCY REGULARIZATION FOR SEMI-SUPERVISED LEARNING

The previous chapter introduced the concept of semi-supervised learning and how it may be beneficial in the limited labeled data scenario. A generative modeling approach for SSL is discussed and applied to the MIC task to address the challenge of missing labels. This approach, however, does not improve performance compared to the weakly supervised attention model for musical instrument classification proposed in Chapter 3 even though the attention model only utilizes labeled information. This chapter focuses on a different approach to semi-supervised learned based on a concept known as *consistency regularization* and aims to answer RQ3 regarding the impact of unlabeled data on MIC models.

5.1 Background

Section 4.1.1 covers a vast majority of methods developed for SSL. Consistency regularization is another SSL technique which has seen a fair amount of success in semi-supervised learning benchmarks. The main concept behind consistency regularization is that a model should output consistent predictions for perturbed versions of the same input. These perturbations vary from simply using dropout in the model architecture [135], to applying various data augmentation functions to the inputs [63, 136, 137]. In this way, models can use unlabeled data during training by penalizing any variability in outputs under input or model perturbations. Often these methods are also known as 'student-teacher' models since the 'student' model compares its output for unlabeled data against a reference or 'teacher'. Consistency regularization methods follow the *clustering assumption* of semi-supervised learning. Under this assumption, two data points belonging to the same cluster in the input space should also belong to the same class. This also implies that a decision boundary drawn in the input space will pass through regions of low density. This is known as the *low density*

separation assumption [138].

Sajjadi et al. proposed the Π model wherein the unlabeled images undergo random cropping and rotations, models contain dropout and randomized max-pooling schemes (for CNNs) [63]. Here, the model compares its own outputs for stochastically perturbed data, i.e. the student is its own teacher. Miyato et al. used a similar approach except they use adversarial perturbation of the data instead of standard data augmentation [139]. Laine and Aila extended the Π model using *temporal ensembling* (TE) where an ensemble of predictions for unlabeled data is maintained during the training process [136]. The ensemble prediction is updated at the end of every epoch using a moving average of past predictions and the predictions from the current model. During training, consistency is measured using the difference of model predictions from the ensemble predictions.

Tarvainen and Valpola developed a method known as *Mean Teacher* (MT) where student model weights are aggregated using a moving average of previous training iterations or steps [62]. The so called 'Mean Teacher' refers to the weighted average model. The main difference between the MT approach and TE model is that model weights are averaged instead of model predictions. MT-based models significantly outperformed other approaches for semi-supervised learning on standard benchmarks. Athiwaratkun et al. modify the MT approach by introducing a different optimization schedule and averaging student weights at different epochs to obtain the teacher model instead of every training iteration [140].

In MIR literature, there has not been much work that focuses on consistency regularization specifically. Wu and Lerch explore the student-teacher paradigm for automatic drum transcription using labeled and unlabeled data [130]. They use two completely different drum transcription models as the student and teacher models and use soft targets — also known as pseudo labels — produced by the teacher for unlabeled data to train the student model. This is similar to consistency regularization since there is a notion of predicting close to a reference model. However, there is no stochastic perturbation of the data or the model parameters which is typical of consistency regularization methods.

5.2 Problem Formulation

In Section 4.2, the multi-label MIC problem is decomposed into 20 binary classification problems: one for each instrument. The binary relevance decomposition leads to a partitioned labeled and unlabeled dataset for each instrument, $\mathbb{D} = \mathbb{D}_l^i \dot{\cup} \mathbb{D}_u^i$, where $\mathbb{D}$ is the full OpenMIC dataset, $\mathbb{D}_l^i$ is the labeled dataset for the ith instrument, and $\mathbb{D}_u^i$ is the unlabeled dataset for the ith dataset. This is favorable for the SSL framework where most research focuses on multi-class problems with disjoint labeled and unlabeled datasets rather than multi-label problems with missing labels. However, the binary relevance decomposition takes away the ability of the instrument-specific models to even implicitly learn any correlations between instruments which may explain the poor performance of DGM as compared to ATT.

For this experiment, the exact same setup as Section 3.2 is used. The MIC task is formulated as a MIML classification problem with missing labels. Additionally, the ATT model architecture is utilized as the underlying classifier for the experiments in this chapter. This enables the proposed method to not only tackle the problem of missing labels but also tackle the problem of weakly labeled data. The main difference between the problem formulation of ATT and MT is the addition of the teacher model and the consistency regularization loss. Contrasted with the DGM method for SSL, the MT model is trained in a multi-label setting.

5.3 Method

This section describes the MT method for SSL in more detail and discusses how it is applicable to the MIC task which is formulated as a MIML problem with missing labels.

5.3.1 Mean Teacher

Formally, the consistency regularization loss can be written as [62]:

$$\mathcal{J}_{\mathcal{CR}} = \mathbb{E}_{\mathbf{x},\eta,\eta'}\left[\mathcal{D}(f(\mathcal{T}(x),\theta,\eta), f(\mathcal{T}(x),\theta',\eta'))\right], \tag{5.1}$$

where $\mathcal{D}$ is some distance function, f represents the computational graph of a deep neural network with parameters θ and stochastic noise (dropout) η, and $\mathcal{T}$ is some data $\mathbf{x}$ transformation function. The network with parameters θ is the student model, and the network with parameters θ' is the teacher model.

In the case of the Π model, $\theta' = \theta$, i.e., the student and teacher are the exact same model. In TE, the teacher weights θ' are not explicitly maintained. Instead, $f(\mathcal{T}(x),\theta',\eta')$ is approximated by ensembling the student outputs after every epoch.

In MT, the teacher weights θ are the exponential moving average (EMA) of the student model weights θ after every training step:

$$\theta'_t = \alpha\theta'_{t-1} + (1-\alpha)\theta_t, \tag{5.2}$$

where α is the EMA weight. During inference or testing, either the student or teacher models may be used although the teacher model typically performs slightly better than the student model due to the ensemble effect of the moving average.

The MT approach has the benefit of producing superior teacher models as compared to the Π or TE models. The main reasons being (i) the use of model weight averaging instead of relying solely on input perturbations, and (ii) the rapid feedback cycle in MT compared to TE. The model weight averaging method in MT lay the foundation for several other SSL approaches [140, 137].

5.4 Experiment

The aim of this experiment is to study the effect of using the unlabeled data in the OpenMIC dataset. To that end, the MT model is compared against the ATT. This is a fair comparison since both models use the same underlying architecture.

5.4.1 Semi-Supervised Loss Function

The main difference between the traditional SSL setup and the MIC setup is the lack of a disjoint set of labeled and unlabeled data in the MIC setup. The MIC setup uses a single dataset which has partially labeled data points. This, however, does not pose a serious problem. Since the MIML formulation treats each instrument label as independent, it is straightforward to treat each input data point as a labeled sample for labeled instrument, and as an unlabeled sample for unlabeled instruments.

The BCE_p loss function (see Section 3.2.2) is used for the labeled part of the data. For the labeled and unlabeled instruments, the consistency loss from equation 5.1 is utilized. The final loss function can be expressed as:

$$\mathcal{J}_{\int\int}(\mathbf{y}, \mathbf{q}, \mathbf{q}') = \text{BCE}_p(\mathbf{y}, \mathbf{q}) + \beta \mathcal{J}_{\mathcal{CR}}(\mathbf{q}, \mathbf{q}'), \tag{5.3}$$

$$\mathcal{J}_{\mathcal{CR}}(\mathbf{q}, \mathbf{q}') = \frac{1}{20}||\mathbf{q} - \mathbf{q}'||^2, \tag{5.4}$$

where $\mathbf{y}$ is the label vector associated with a data point, $\mathbf{q}$ is the student prediction, $\mathbf{q}'$ is the teacher prediction, and β is a hyperparameter which decides the relative weight of the two loss terms. Note that $\mathbf{y}$ consists of both observed and unobserved or missing instrument labels. The BCE_p term takes into account only the observed labels and is the supervised component of the loss function, while the $\mathcal{J}_{\mathcal{CR}}$ takes into account both the labeled and the unlabeled instruments and is the unsupervised component of the loss function. Thus, the MT model is capable of leveraging both the labeled and unlabeled data for MIC.

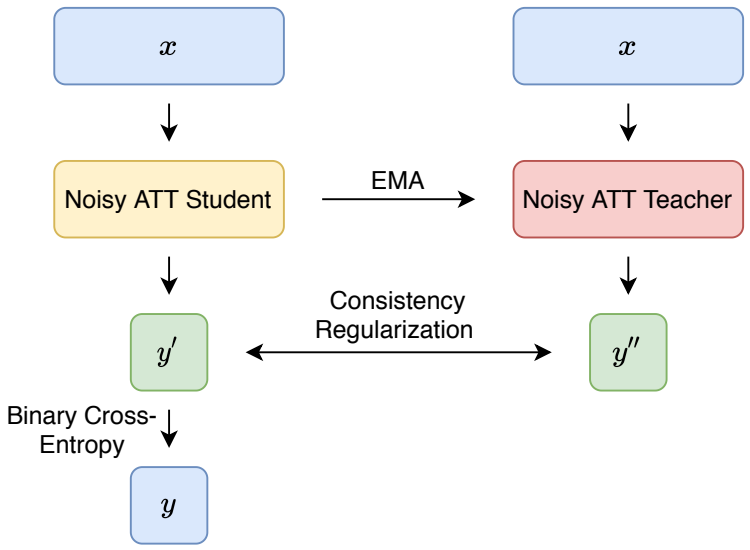

Figure 5.1: The Mean Teacher model training involves two copies of the ATT model: a student and a teacher. Both are labeled as noisy they utilize dropout regularization and therefore produce stochastic outputs during training. The student model performs model parameter updates using a gradient descent-based optimization algorithm (Adam), while the teacher model does not update itself using gradient descent. Instead the teacher model parameters are calculated as an exponential moving average of student model parameters at each step of the training process. The student model loss is computed using binary cross-entropy for the labeled part of the OpenMIC dataset. An additional consistency regularization term is added to penalize inconsistency between the student and teacher outputs.

5.4.2 Training Procedure

As in previous experiments, VGGish features are used as inputs to the models. No input transformation is used, instead relying solely on dropout in the model architecture of the ATT model to introduce stochasticity. Figure 5.1 shows the high-level flow-chart of the MT training process. The model architecture of the underlying ATT model is exactly the same as that used in Section 3, enabling a fair comparison of the supervised and semi-supervised approaches. The EMA weight α is set to 0.999.

The MT model is trained for 200 epochs using the Adam optimizer [106]. The learning rate used is 0.001 and the momentum terms β_1 and β_2 are set to 0.9 and 0.999, respectively. Additionally, the learning rate is decayed by half every 30 epochs. The unsupervised loss weight β is set to 3 after performing a grid search. An exponential ramp-up function is

Table 5.1: Overall results for MT versus supervised-only ATT. Overall results for DGM versus supervised-only ATT and FC models. The mean and standard deviation of the results are shown across 5 seeds. Metrics are shown in percentage. Differences are statistically significant.

	Overall Precision	Overall Recall	Overall F1-score	AUROC
ATT	81.65 ± 0.18	80.68 ± 0.17	80.84 ± 0.18	89.08 ± 0.08
MT	$\mathbf{82.13 \pm 0.10}$	$\mathbf{80.93 \pm 0.09}$	$\mathbf{81.26 \pm 0.05}$	$\mathbf{89.33 \pm 0.03}$

applied to β such that it ramps up to the maximum value in 100 epochs. The ramp function is specified in Appendix A.

For consistent and fair evaluation, I use the exact same splits for training and testing as in sections 3 and 4. Additionally a validation set consisting of 15% of the training set is used. The teacher model with the lowest validation loss is finally used for testing. All experiments are run with 5 different random seeds.

5.4.3 Results and Discussion

Table 5.1 and Figure 5.2 show the overall results, and the instrument-wise results, respectively. The MT model outperforms the ATT slightly in terms of overall performance, and for a majority of instruments.

Comparing the overall performance of the two methods, the MT model marginally outperforms the ATT model. While the improvement is not considerable, the differences are statistically significant. Additionally, while the performance of the two models for several instruments is almost the same, the MT model outperforms the ATT model for most of the instruments where performance is poor, such as clarinet, accordion, and flute. In the ATT case, which is a noisy model without consistency regularization, the model may output vastly different predictions for the same input data for the poor performing instruments. In the MT model, the teacher can be treated as a smoothed ensemble model, thus producing relatively stable outputs for difficult labels. Penalizing the difference between the student and teacher model this leads to improved performance.

Tarvainen and Valpola showed that the MT model typically outperforms fully supervised

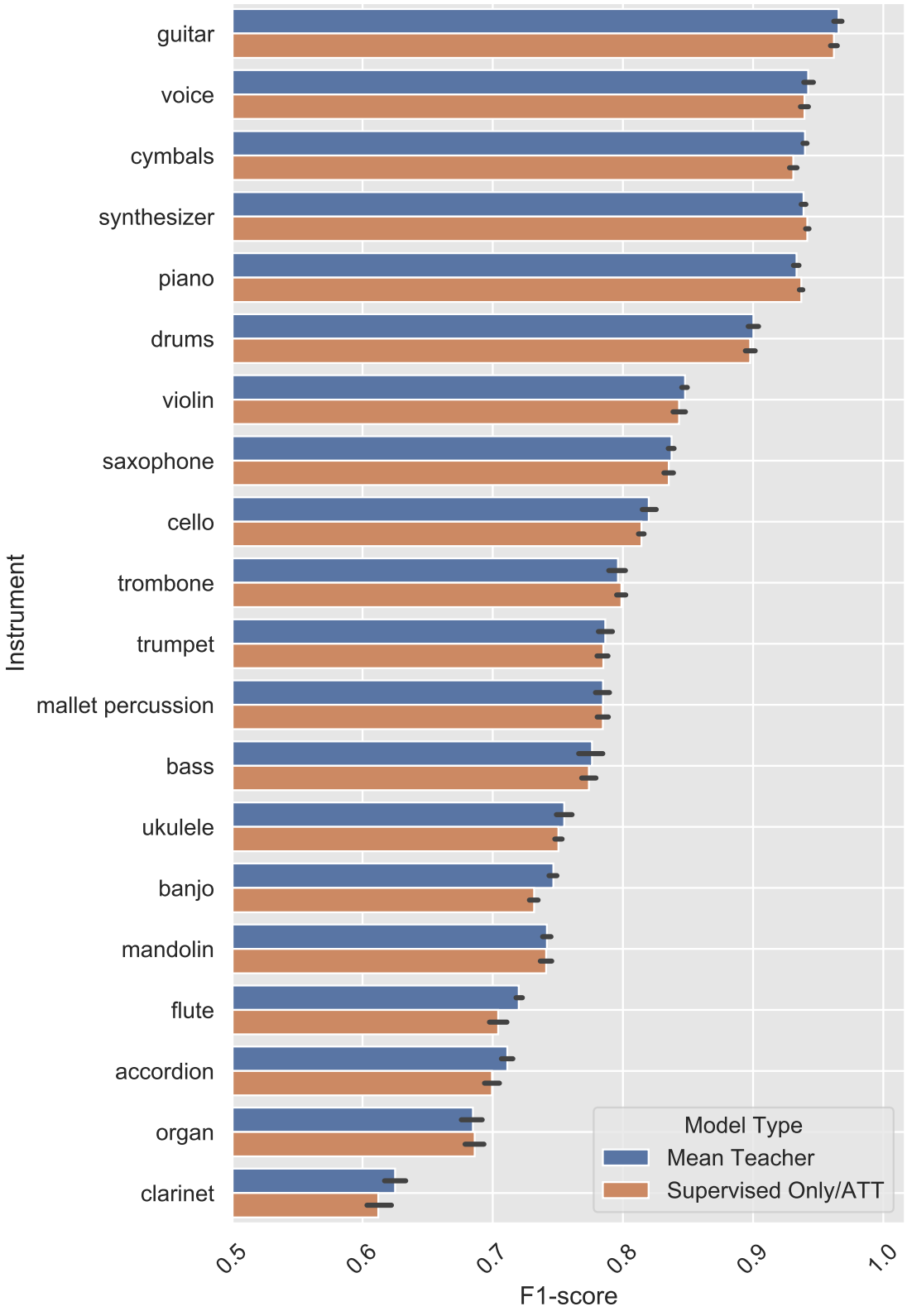

Figure 5.2: Macro-avg F1-scores for all instruments comparing the MT model and supervised-only ATT model.

models trained on equal amounts of labeled data using multi-class image datasets such as CIFAR10 and SVHN [62]. However, there are a few variables that they control during these experiments:

(i) Number of labeled examples (L)

(ii) Number of unlabeled examples (U)

(iii) Class balance in both labeled and unlabeled data

Typically, $U = |\mathbb{D}| - L$, where $\mathbb{D}$ is the dataset being used. The labeled and unlabeled datasets are created by randomly removing labels from a fixed number of data points in the dataset. Additionally, both the labeled and unlabeled datasets have class balance. Most experiments use labeled data ratios $\frac{L}{|\mathbb{D}|}$ ranging between 0.3% to 10%. The smaller the ratio, the larger the gap between MT and the fully supervised baseline.

In the case of MIC, however, the OpenMIC dataset is utilizes as is. The OpenMIC training dataset has approximately 30000 out of around 300000 labels. Unlike image datasets, this does not imply that there are 270000 unlabeled data points. The MIC task is multi-label instead of multi-class, implying that the same data point may be associated with multiple labels. Hence, the total number of data points is much smaller, at around 15000 training audio clips compared to 50000 images in CIFAR10 and close to 75000 images in SVHN. The fact that MIC is multi-label also makes it unclear how to calculate the ratio of labeled to unlabeled data. In the image classification benchmarks there is a clear separation between labeled and unlabeled images, while in the OpenMIC dataset each audio clip is partially labeled and has at least 1 instrument label. Considering $L \approx 30000$, the labeled data ratio is $\approx 10\%$ which is on the higher end of labeled data ratios used in semi-supervised image classification experiments, which may explain the small difference in performance between the MT and ATT models.

Additionally, there are many instruments in the MIC task with class imbalance. The use of consistency regularization with class imbalance may lead to additional difficulties in

the learning process due to the clustering assumption. Consistency regularization typically pushes the decision boundary of classifiers towards regions of low density in the latent space. Class imbalance may lead to additional difficulties causing the decision boundary to traverse through high density regions of the minority class [141].

Controlled experiments to understand the behavior of MT are difficult using the OpenMIC dataset, due to the aforementioned reasons. To better understand the behavior of MT for music classification, an ablation study can be performed using the closely related task of automatic music tagging. Music tagging is one of the few music classification tasks with a large-scale dataset available: the Million Song Dataset (MSD)[58]. The goal of the following experiment is to determine the effectiveness of the MT approach for music classification using a large-scale dataset. The benefit of using a large-scale dataset is that controlled experiments with varying labeled data ratios can be conducted. In this case, unlike OpenMIC, data points are either labeled or completely unlabeled, implying that the labeled and unlabeled datasets are disjoint instead of overlapping.

5.5 Varying Labeled Data Ratio

Bertin-Mahieux et al. released the MSD to encourage commercial scale MIR research [58]. The MSD is a collection of approximately one million song features along with various tags obtained from Last.fm[1]. These tags contain information regarding genre, instrumentation, era, tempo, etc. Similar to the MIC task, the music tagging task is a multi-label classification problem. The music tagging task is also setup as a weakly labeled audio classification task, similar to the MIC task.

The main difference in experiment setup between this music tagging experiment and the MIC experiment in Section 5.4 is the scale of data available for experiments. The MSD enables experiment setup identical to research in semi-supervised image classification where large datasets are partitioned into labeled and unlabeled subsets. This facilitates varying

[1]https://www.last.fm (Last accessed 7/8/2020)

Table 5.2: The ATT model compared to two state-of-the-art (SOTA) models for music tagging using the macro-average AUROC metric. All the datasets are trained with the MSD using identical data splits. The performance of the SOTA models is obtained from a large-scale evaluation of music tagging models performed by Won et al. [142].

	AUROC
ATT	87.48
Harmonic CNN SOTA [143]	88.98
CRNN SOTA [31]	84.60

the labeled data ratio as well. The OpenMIC dataset on the other is already a smaller scale dataset and further reducing the amount of labels will likely lead to poor performance compared to supervised-only approaches.

5.5.1 Pre-processing

Following established literature on music tagging, a subset of ≈ 240000 29 s song clips are selected from the MSD as these are tagged with the 50 most frequent tags. A standard split[2] for training, testing, and validation is used for music tagging with MSD.

To stay as close to the experiment settings used for MIC, VGGish features are used for the tagging task as well. The ATT model is chosen as the underlying model architecture. To validate the choice of architecture, the ATT model using labeled data is compared against standard models from music tagging literature. Table 5.2 shows that the ATT model achieves close to state-of-the-art performance on music tagging, making it a viable underlying architecture for the MT model.

5.5.2 Semi-Supervised Music Tagging

In the OpenMIC dataset, there is no separate labeled and unlabeled dataset due to each data point being partially labeled. For this experiment, however, the MSD training set $\mathbb{D}(|\mathbb{D}| \approx 200000)$ is divided into two disjoint subsets $\mathbb{D} = \mathbb{D}_l \dot{\bigcup} \mathbb{D}_u$: a labeled subset $\mathbb{D}_l$ and unlabeled subset $\mathbb{D}_u$. To achieve this, data points are randomly selected from $\mathbb{D}$ and added to

[2] https://github.com/keunwoochoi/MSD_split_for_tagging (Last accessed 7/8/2020)

$\mathbb{D}_u$ without the labels. This is similar to how experiments are carried out for image datasets, as described in Section 5.4.3. This process gives full control over the amount of labeled and unlabeled data during the experiments. It is important to note that the class imbalance problem still exists in the music tagging task unlike the image classification experiments.

Experiments are carried out with different ratios $|\mathbb{D}_l|/|\mathbb{D}|$ selected from:

$$|\mathbb{D}_l|/|\mathbb{D}| \in \{0.01, 0.02, 0.04, 0.08, 0.16, 0.32\}. \tag{5.5}$$

For each labeled data ratio, the MT model is trained using the remaining partition of the dataset as unlabeled data. For comparison, ATT models are also trained for each labeled data ratio. For $|\mathbb{D}_l|/|\mathbb{D}| = 0.08$ the labeled data ratio is close to that of the OpenMIC dataset.

Training Procedure

For this experiment, pairs of semi-supervised MT and supervised-only ATT models are trained under different labeled data ratios.

For the MT models, the underlying model architecture is almost identical to the ATT model from Section 3. The only change made was using a dropout with probability 0.5 instead of 0.6. A grid search is performed for different values of the unsupervised loss weight $\beta \in 16, 32, 64$. The EMA weight $\alpha = 0.999$ is the same as the MIC experiment. The model is trained for 100 epochs of the unlabeled subset using the Adam optimization algorithm [106], with the same initial hyperparameters as the MIC experiment (see Section 5.4.2). Note that in one training step two batches of data are processed through the model: first, the batch sampled from the labeled subset, and second, the batch sampled from the unlabeled subset. The learning rate schedule, however, is different. Instead of using a constant decay after a fixed number of epochs, in this case I use exponential ramp-up and ramp-down functions for the learning rate. A similar schedule is used by Tarvainen and Valpola [62]. The same ramp-up function is also used for the β term. The ramp-up is

performed for 10 epochs and ramp-down is performed for 50 epochs. Details of the ramp-up and ramp-down functions can be found in Appendix A. The teacher with the best validation loss is chosen for testing.

The supervised-only ATT models are also trained using the Adam optimizer for 200 epochs of only the labeled subset using the same initial optimizer hyperparameters as above. Technically, training for 200 epochs would be unfair since the effective number of labeled data epochs in the MT models is much higher. However, it was observed that models typically converge within 200 epochs in this case. Learning rate is ramped-up and ramped-down using the same functions as discussed previously. Ramp-up is performed for 20 epochs and ramp-down is performed for 100 epochs. As before, the model with the best validation loss is chosen for testing.

All experiments are run three times with different random seeds.

5.5.3 Results and Discussion

Figure 5.3 compares the performance of the MT model against the supervised ATT model. It is very clear from the plot that the amount of labeled data has a very strong influence over the relative performance of the two models. After a certain amount of labeled data, the fully supervised model is able to match the performance of the semi-supervised model.

An interesting observation is that while the performance of MT with a small labeled data ratio is typically better than the fully supervised model with the same ratio, the MT model is typically inferior to a fully supervised model with double the amount of labeled data.

These results point to possibility that the OpenMIC dataset contains just enough labels such that the fully supervised ATT model performs only slightly worse than the semi-supervised MT model.

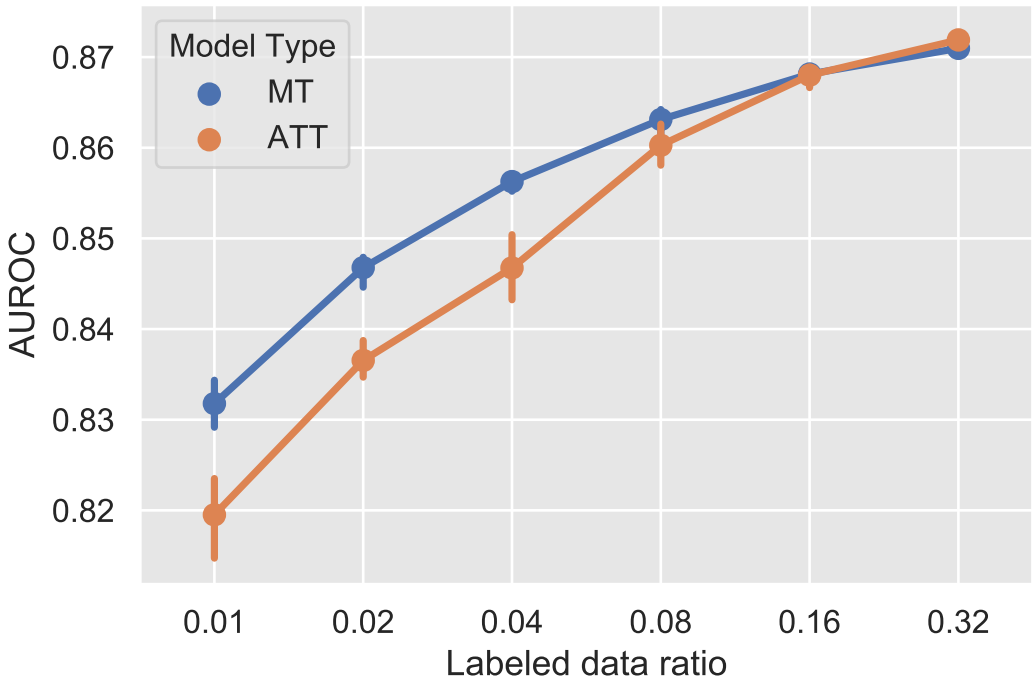

Figure 5.3: AUROC on the MSD test set using different amounts of labeled data. It is clear that the MT model is especially effective when very few labels are used. Another observation that can be made is that the supervised-only model benefits more strongly from inclusion of additional labeled data.

5.6 Conclusion

The consistency regularization approach fares better than the generative modeling approach for SSL. One of the major differences between the two semi-supervised MIC setups was that the consistency regularization approach simply modified an already well performing approach — the ATT model — to also take into account unlabeled data. As discussed in Section 4.5.3, the ATT model benefits from inductive biases which give it the upper hand in dealing with weakly labeled data.

Although the difference between the overall performance of the MT model against the ATT is not very high, there is a statistically significant improvement. Similar improvements are observed for the music tagging task with a larger scale dataset. This is encouraging since it demonstrates that methods that leverage unlabeled data are applicable to the MIC task as well as to other MIR tasks. However, the results of the experiments with the MSD also demonstrate that SSL methods are most beneficial when there are very few labels available and even then, they are unable to outperform fully supervised methods with access to even slightly more labeled data.

Finally, the results for MT also demonstrate low variability across different random initialization and training/validation splits. The low variability can be attributed to the consistency regularization as well as the model weight averaging process, which essentially works as a low pass filter over the student model weight updates. This insight can also be shared across other MIR tasks that deal with deep neural networks.

CHAPTER 6
CONCLUSION

Research on MIC has largely relied on the availability of labeled training data. This reliance on annotated data extends to other MIR tasks as well. With the advent of deep learning, the need for labeled data has increased as deep learning-based models continue to be adopted in MIR research. However, obtaining strongly labeled data at a large scale is both difficult and costly. Weakly labeled data, on the other hand, is comparatively easier to obtain. The challenge lies in the fact that existing methods for MIC are geared towards finely labeled short clips of music, whereas weakly labeled data tend to be longer clips of coarsely labeled music. These *strongly supervised* methods are found to perform poorly when directly applied to weakly labeled data as shown in Section 3.3. Weakly labeled data also tend to have incomplete labeling which poses an additional challenge. Thus, I present different strategies to leverage weakly labeled data for the task of MIC.

These strategies may be classified into two categories depending on the problem they aim to solve: (i) learning effectively from weak labels in music data, and (ii) leveraging missing labels or unlabeled music data. The method presented in Chapter 3 belongs to the first category, the method presented in Chapter 4 belongs to the second category, and the method presented in Chapter 5 belongs to both categories. These methods share the common property of being 'weakly supervised', either by virtue of using weakly labeled data for training, or by virtue of utilizing missing labels or unlabeled data.

Chapter 3 answers RQ1 by formulating the MIC task as a MIML problem. This framework enables the use of an attention-based prediction aggregation strategy that outperforms models based on previous state-of-the-art approaches for MIC. The attention mechanism works well for WLD since it simplifies the task by attending to parts of the input which are relevant for a particular instrument, thus separating the task of localization and detection.

As discussed in Section 3.4, this approach makes a few simplifying assumptions which may not hold for music. In this model, every instance is assumed to be independent from other instances. This assumption does not necessarily hold for music, where neighboring instances in a song will likely be highly correlated. Hence a possible extension of the approach involves relaxing the instance independence assumption. Additionally, in the MIML framework, all the labels are considered independent from each other. Again, this assumption is very strict for musical instruments. While the model may implicitly learn instrument correlations due to the shared embedding layers, it may be worth exploring methods that explicitly model label correlations.

Chapter 4 jointly answers RQ2 and RQ3. In a paradigm shift from purely discriminative modeling, generative modeling is applied to the task of MIC with missing labels. The binary relevance decomposition is applied to the OpenMIC dataset to convert the multi-label problem into multiple binary classification problems, thus dividing the dataset into a labeled set and an unlabeled set for each instrument. A semi-supervised DGM is presented which models the generative process of the data as a function of latent factors and the labels. This model outperforms supervised-only baseline but fails to outperform the supervised-only ATT model. The poorer performance compared to the attention model may be explained by the fact that the generative model is not equipped with mechanisms to deal with weakly labeled data. The input audio is generally longer (10 s) than typical generative models of audio are trained with. Additionally, ATT has the benefit of being trained in a multi-label setting. Incorporating these inductive biases into the DGM may lead to further improvements.

Chapter 5 answers RQ3 by following a different method for SSL based on consistency regularization. This approach relies on penalizing inconsistency in model outputs for stochastically perturbed data. This regularization method is unsupervised and hence may be applied to both labeled and unlabeled data. The particular method, called the Mean Teacher, utilizes a student-teacher approach. The MT model outperforms even the ATT model. Although the difference in performance is not large, the improvement is encouraging and

Table 6.1: Characteristics of the three proposed models for MIC

	For Weakly Labeled	Semi-Supervised	Multi-Label	Generative
ATT	✓	✗	✓	✗
DGM	✗	✓	✗	✓
MT	✓	✓	✓	✗

Table 6.2: Comparison of all proposed models in terms of overall metrics. Metrics are shown as percentages. The mean and standard deviation is shown across different random seeds.

	Overall Precision	Overall Recall	Overall F1-score	AUROC
ATT	81.65 ± 0.18	80.68 ± 0.17	80.84 ± 0.18	89.08 ± 0.08
DGM	79.79 ± 0.16	81.14 ± 0.11	79.54 ± 0.19	88.78 ± 0.05
MT	82.13 ± 0.10	80.93 ± 0.09	81.26 ± 0.05	89.33 ± 0.03

shows that unlabeled data can indeed be used in a limited labeled data scenario. Additional experiments carried out with the larger scale MSD for music tagging reinforce the validity of these claims (see 5.5). The MT approach shows a considerable improvement over supervised-only models for very small labeled data ratios.

6.1 Contributions

The main contribution of this thesis is the introduction of novel methods for MIC which can effectively learn using limited labeled data. These methods also establish new state-of-the-art performance for MIC on the OpenMIC dataset (see Table 6.2 and Table 6.1). Literature in MIC — and MIR in general — rarely discusses learning with limited amounts of labeled data and hence this thesis adds to the repertoire of techniques that the community can utilize to make the most of available data.

6.1.1 Multi-Instance Learning and Attention for MIC

The first contribution of this thesis is an attention-based model developed to handle weakly labeled data. The MIC task is formulated as a MIML problem wherein the input audio clip is treated as a bag of multiple instances. The labels for the clip are known but individual instance labels are unknown. The proposed model independently predicts the instance-level

labels and uses an attention-based aggregation strategy to obtain the clip-level labels. In this manner, the model has the ability to detect presence of instruments in long clips of music regardless of the duration of their activity. The model is compared against various methods for MIC that are not specifically designed for weakly labeled data, as well as naive prediction aggregation methods. The proposed method outperforms all baseline methods.

6.1.2 Semi-Supervised Deep Generative Model

The second contribution is a novel generative modeling-based system for MIC that is capable of leveraging missing labels in the OpenMIC dataset. This is the first model for MIC that combines SSL and generative modeling to learn from both labeled and unlabeled data. The semi-supervised DGM models the data generative process using latent factors and the class label as well, thus allowing labels to be incorporated in the typically unsupervised VAE learning process [61, 60]. The model additionally learns a classification network in tandem that infers labels for unlabeled data. The proposed model is compared against a simple supervised classification network and the weakly supervised attention model, both of which are only trained using the labeled part of the dataset. The generative model outperforms the simpler classification network but fails to outperform the attention model.

6.1.3 Mean Teacher Model

The final contribution of the thesis is a second SSL-based approach: the MT model. The MT model is a consistency regularization-based SSL technique that utilizes stochasticity in the network and data, as well as model weight averaging. The consistency regularization loss is an unsupervised loss and hence can be used for the missing labels in the OpenMIC dataset. In the proposed MT approach, the same input is provided to a student and teacher model, both of which use dropout as a source of stochasticity. The teacher model is the exponential moving average of student model weights from all previous training steps. The unsupervised loss term encourages the student to produce outputs ‘consistent’ with the teacher model.

This method marginally outperforms even the attention-based model.

6.1.4 Additional Contributions

In the early stages of this work, MIC using strongly labeled data is explored. The main focus was to evaluate different categories of neural networks applied to the IAD task.

To that end, the MedleyDB [78] multi-track dataset is extended by collecting, and processing 258 additional multi-track data, released as the Mixing Secrets [79] dataset, with broad impact in both MIC as well as music source separation research. Subsequently MLPs, CNNs, and CRNNs are trained to recognize 18 instruments in short 1 s clips of audio. These models are evaluated at various time-resolutions. CNNs and CRNNs were found to significantly outperform MLPs. The difference between CNNs and CRNNs was found to be negligible, possibly owing to the short duration of the input samples which render CNNs as effective as the CRNNs in learning temporal features.

Reflecting on the experimental setup, however, pointed to several issues in the data being used to train and evaluate the MIC systems. The multi-track data used to train the systems was severely imbalanced in terms of both instrument classes and genres. Additionally, the total number of unique artists and tracks in the dataset is very low and may cause generalization issues in models trained with this data.

6.2 Future Directions

The methods presented in this thesis explore problems that have not received much attention in MIC research. These pertain to challenges that exist specifically in weakly labeled data: (i) coarse labels for relatively long clips of audio, and (ii) incomplete labeling or missing labels. These are unexplored problems not just for the MIC task but also in the MIR community. The goal of this thesis is to guide future research towards solving these data challenges by adopting methods for semi-supervised learning or generative modeling. Having set the foundation and establishing the effectiveness of these methods, I propose a

few research directions that may lead to more breakthroughs in the MIC task.

6.2.1 Transfer Learning

All the proposed methods in this thesis share the same transfer learned VGGish features. While transfer learning has been shown to perform well in most contexts, in this case the domain of data used to train the transfer learning VGGish model may be too general. The VGGish CNN is trained using an audio dataset extracted from Youtube videos at the scale of multi-million 10 s audio clips. While music does make up a part of the dataset, it also contains other verticals such as video games, food, vehicles, etc. The use of a model trained with out-of-domain audio may limit the advantage of using the transfer learned features directly. Music has temporal and harmonic structure often absent in other modalities of audio such as speech or environmental audio recordings.

A straightforward solution to this is to fine-tune the VGGish model using the OpenMIC dataset instead of using it as a feature extractor. Another approach to use models trained with a larger music dataset which may be more capable of extracting music-specific features. These models may be trained in the supervised manner for tasks accompanied by large datasets such as the MSD and subsequently used as feature extractors. An alternative is to use unsupervised representation learning methods with large datasets. Another approach to tackle the problem of domain shift is domain adaptation.

6.2.2 Relaxing MIL Assumptions

The MIL framework is utilized in sections 3 and 5 to improve model performance in learning from weakly labeled data. Section 6.1.1 mentions the assumptions in the MIL setting, and that these assumptions do not necessarily hold for music data. One of these is the instance independence assumption. This may easily be relaxed by modifying the instance-level scoring function (see Section 3.2.1). The use of RNNs is already discussed in Section 3. Newer model architectures such as transformers [144] are shown to be able to model self-

similarity in sequential data. These models may also be utilized in the MIL setting to relax the instance independence assumption. Note that this applies to the supervised ATT model as well as the semi-supervised ATT model.

The other assumption is that of label independence. Models strictly following the MIML framework will ignore any relationships between labels. The ATT model shares weights in the embedding layer and is therefore able to learn a shared representation for all labels. However, methods that explicitly model label correlations have shown to significantly improve multi-label classification systems, thus making it a viable research direction for the future.

6.2.3 Data Augmentation in Consistency Regularized SSL

The ATT method presented in Section 5 relies on dropout in the model architecture as the sole source of stochasticity. In recent SSL literature, the benefits of using weak and strong image data augmentation methods have been illustrated [116, 145]. In images, weak augmentation refers to simple transformations such as translation, rotation, and cropping. Strong augmentations are involve heavy distortions to the source image. Data augmentation strategies have also been studied for music [146, 147]. These may be leveraged in the ATT setting directly or in other consistency regularization-based SSL methods.

I conclude my thesis with the hope that future research in MIC can gain inspiration from the introduction of methods aiming to solve problems that are yet to be explored in the literature. These include:

1. Weakly labeled MIC: an attention mechanism to solve the weakly labeled MIC task framed as a MIL problem is proposed.

2. Learning from limited labeled data: generative modeling and consistency regularization approaches for SSL are presented in this thesis.

Stepping back and focusing on the larger goal of improving MIR systems, this thesis brings

to light various data challenges that exist in a field that is increasingly reliant on data-driven methods and recommends different ways of addressing these challenges.

Appendices

APPENDIX A
RAMP FUNCTIONS FOR MT TRAINING

A.1 Ramp-up

A sigmoid-shaped ramp-up function is used as described by Tarvainen and Valpola [62]:

$$w(x) = \exp -5(1-x)^2, \tag{A.1}$$

where $x \in [0, 1]$. As x goes linearly from 0 to 1 (number of epochs reach the ramp-up termination criteria), $w(x)$ follows the curve shown in Figure A.1.

A.2 Ramp-down

Ramp-downs also follow the sigmoid shape:

$$w(x) = 1 - \exp -12.5x^2, \tag{A.2}$$

where $x \in [0, 1]$. As x goes linearly from 0 to 1 (number of epochs reach the ramp-down termination criteria), $w(x)$ follows the curve shown in Figure A.2.

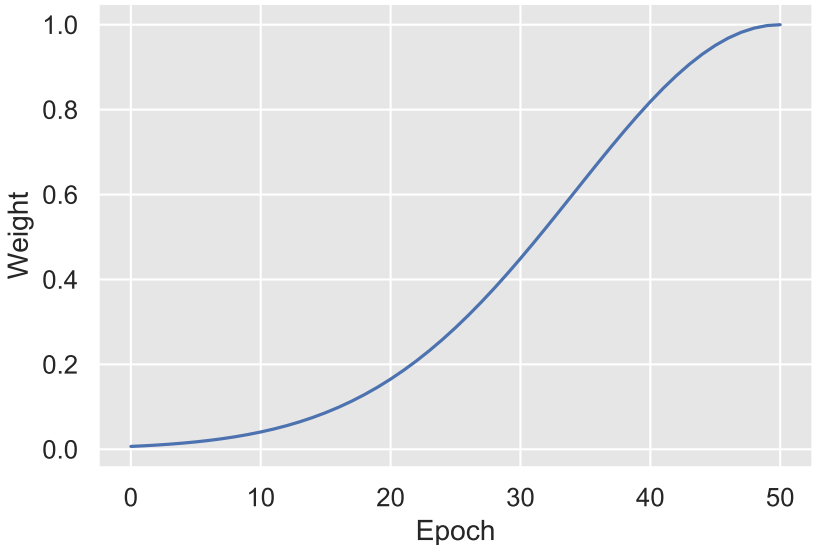

Figure A.1: Ramp-up function used during mean teacher training. Here the ramp-up happens for 50 epochs

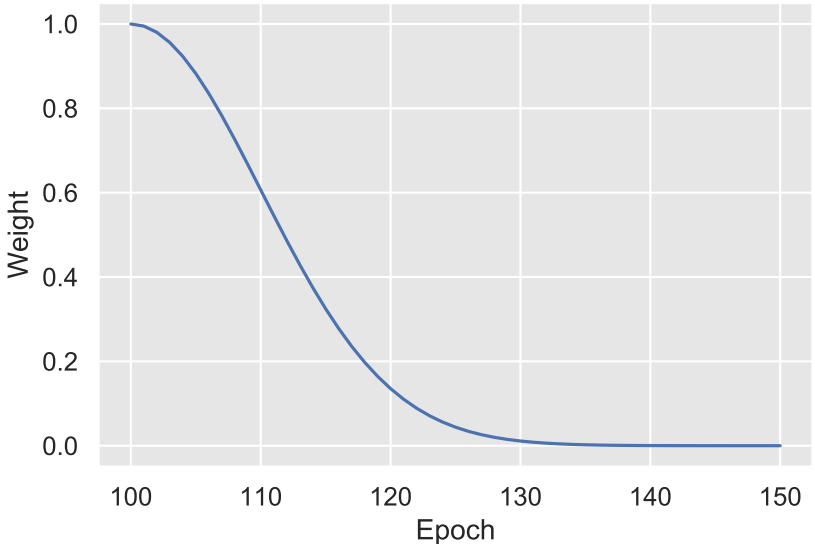

Figure A.2: Ramp-down function used during mean teacher training. Here the ramp-down happens for 50 epochs

APPENDIX B

TRANSFER LEARNING WITH VGGISH

All experiments performed with the OpenMIC dataset in this thesis used the VGGish embeddings that accompany the dataset. These embeddings are extracted from a deep CNN architecture very similar to VGG-11 architecture. The VGGish model uses 4 convolutional blocks instead of 5 and uses a 128-dimensional fully connected layer at the end. The VGGish model is trained using large amounts (more than 8 million) of Youtube audio clips for audio classification.

The OpenMIC dataset was passed through the pre-trained model to obtain the VGGish features. Each clip in the dataset represented by a 10×128-dimensional feature. Each embedding is composed of 128 8-bit integers obtained for a 0.96 s audio clip as input to the model.

REFERENCES

[1] P. Hill, "From Score to Sound," in *Musical Performance – A Guide to Understanding*, J. Rink, Ed. Cambridge: Cambridge University Press, 2002, pp. 129–143, ISBN: 978-0-521-78862-5.

[2] C. E. Seashore, *Psychology of Music*. New York: McGraw-Hill, 1938.

[3] J. A. Grahn and M. Brett, "Rhythm and Beat Perception in Motor Areas of the Brain," *Journal of cognitive neuroscience*, vol. 19, no. 5, pp. 893–906, 2007.

[4] Y. Nan, T. R. Knösche, and A. D. Friederici, "Non-Musicians' Perception of Phrase Boundaries in Music: A Cross-Cultural ERP Study," *Biological Psychology*, vol. 82, no. 1, pp. 70–81, 2009.

[5] P. Iverson, "Auditory Stream Segregation by Musical Timbre: Effects of Static and Dynamic Acoustic Attributes," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 21, no. 4, pp. 751–763, 1995.

[6] American National Standards Institute. and Acoustical Society of America., *American National Standard: Acoustical Terminology*, ser. Acoustical Terminology. New York: Standards Secretariat, Acoustical Society of America, 1994.

[7] J. M. Grey and J. W. Gordon, "Perceptual Effects of Spectral Modifications on Musical Timbres," *The Journal of the Acoustical Society of America*, vol. 63, no. 5, pp. 1493–1500, 1978.

[8] S. McAdams, J. W. Beauchamp, and S. Meneguzzi, "Discrimination of Musical Instrument Sounds Resynthesized with Simplified Spectrotemporal Parameters," *The Journal of the Acoustical Society of America*, vol. 105, no. 2, pp. 882–897, 1999.

[9] R. D. Patterson, "The Sound of a Sinusoid: Time-interval Models," *The Journal of the Acoustical Society of America*, vol. 96, no. 3, pp. 1419–1428, 1994.

[10] G. R. Charbonneau, "Timbre and the Perceptual Effects of Three Types of Data Reduction," *Computer Music Journal*, vol. 5, no. 2, pp. 10–19, 1981.

[11] J. M. Grey and J. A. Moorer, "Perceptual Evaluations of Synthesized Musical Instrument Tones," *The Journal of the Acoustical Society of America*, vol. 62, no. 2, pp. 454–462, 1977.

[12] E. L. Saldanha and J. F. Corso, "Timbre Cues and the Identification of Musical Instruments," *The Journal of the Acoustical Society of America*, vol. 36, no. 11, pp. 2021–2026, 1964.

[13] J. M. Grey, "Multidimensional Perceptual Scaling of Musical Timbres," *The Journal of the Acoustical Society of America*, vol. 61, no. 5, pp. 1270–1277, 1977.

[14] I. Kaminsky and A. Materka, "Automatic Source Identification of Monophonic Musical Instrument Sounds," in *Proceedings of the International Conference on Neural Networks (ICNN)*, Perth, WA, Australia, 1995, pp. 189–194.

[15] K. D. Martin, "Toward Automatic Sound Source Recognition: Identifying Musical Instruments," *NATO Computational Hearing Advanced Study Institute*, 1998.

[16] J. C. Brown, "Computer Identification of Musical Instruments Using Pattern Recognition with Cepstral Coefficients as Features," *The Journal of the Acoustical Society of America*, vol. 105, no. 3, pp. 1933–1941, 1999.

[17] J. Marques and P. J. Moreno, "A Study of Musical Instrument Classification Using Gaussian Mixture Models and Support Vector Machines," Compaq Corporation, Cambridge Research laboratory, Tech. Rep., 1999.

[18] Y. Han, J. Kim, and K. Lee, "Deep Convolutional Neural Networks for Predominant Instrument recognition in Polyphonic Music," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 25, no. 1, pp. 208–221, 2017.

[19] Y.-N. Hung and Y.-H. Yang, "Frame-level Instrument Recognition by Timbre and Pitch," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018, pp. 135–142.

[20] S. Gururani, C. Summers, and A. Lerch, "Instrument Activity Detection in Polyphonic Music using Deep Neural Networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018, pp. 569–576.

[21] S. Gururani, M. Sharma, and A. Lerch, "An Attention Mechanism for Music Instrument Recognition," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, 2019, pp. 83–90.

[22] P. Li, J. Qian, and T. Wang, *Automatic Instrument Recognition in Polyphonic Music Using Convolutional Neural Networks*, 2015. arXiv: `1511.05520 [cs.SD]`.

[23] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-Based Collaborative Filtering Recommendation Algorithms," in *Proceedings of the International Conference on World Wide Web (WWW)*, Hong Kong, Hong Kong, 2001, pp. 285–295.

[24] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative Filtering Recommender Systems," in *The Adaptive Web: Methods and Strategies of Web Personalization*, ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2007, pp. 291–324, ISBN: 978-3-540-72079-9.

[25] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and Metrics for Cold-Start Recommendations," in *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '02, Tampere, Finland, 2002, pp. 253–260.

[26] B. McFee, L. Barrington, and G. Lanckriet, "Learning Content Similarity for Music Recommendation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 8, pp. 2207–2218, 2012.

[27] J. Schluter and C. Osendorfer, "Music Similarity Estimation with the Mean-Covariance Restricted Boltzmann Machine," in *Proceedings of the International Conference on Machine Learning and Applications and Workshops (ICMLA)*, vol. 2, Honolulu, HI, USA, 2011, pp. 118–123.

[28] D. Wang, S. Deng, X. Zhang, and G. Xu, "Learning Music Embedding with Metadata for Context Aware Recommendation," in *Proceedings of the ACM on International Conference on Multimedia Retrieval (ICMR)*, New York, NY, USA, 2016, pp. 249–253.

[29] W. T. Glaser, T. B. Westergren, J. P. Stearns, and J. M. Kraft, "Consumer Item Matching Method and System," pat. US7003515B1, 2006.

[30] J. Pons, O. Nieto, M. Prockup, E. M. Schmidt, A. F. Ehmann, and X. Serra, "End-to-end Learning for Music Audio Tagging at Scale," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018, pp. 637–644.

[31] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, USA, 2017, pp. 2392–2396.

[32] S. Park, T. Kim, K. Lee, and N. Kwak, "Music Source Separation Using Stacked Hourglass Networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018, pp. 289–296.

[33] D. Stoller, S. Ewert, and S. Dixon, "Wave-U-Net: A Multi-Scale Neural Network for End-to-End Audio Source Separation," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018, pp. 334–340.

[34] F.-R. Stöter, S. Uhlich, A. Liutkus, and Y. Mitsufuji, "Open-Unmix - A Reference Implementation for Music Source Separation," *Journal of Open Source Software*, vol. 4, no. 41, p. 1667, 2019.

[35] R. Hennequin, B. David, and R. Badeau, "Score Informed Audio Source Separation Using a Parametric Model of Non-Negative Spectrogram," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, 2011, pp. 45–48.

[36] S. Ewert, B. Pardo, M. Muller, and M. D. Plumbley, "Score-Informed Source Separation for Musical Audio Recordings: An overview," *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 116–124, 2014.

[37] Z. Duan and B. Pardo, "Soundprism: An Online System for Score-Informed Source Separation of Music Audio," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1205–1215, 2011.

[38] M. Miron, J. J. Carabias-Orti, J. J. Bosch, E. Gómez, and J. Janer, "Score-Informed Source Separation for Multichannel Orchestral Recordings," *Journal of Electrical and Computer Engineering*, vol. 2016, e8363507, 2016.

[39] Y.-N. Hung and A. Lerch, "Multi-Task Learning for Instrument Activation Aware Music Source Separation," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Montreal, Canada, Jan. 1, 2020, published.

[40] R. V. Swaminathan and A. Lerch, "Improving Singing Voice Separation Using Attribute-Aware Deep Network," in *2019 International Workshop on Multilayer Music Representation and Processing (MMRP)*, Milano, Italy, 2019, pp. 60–65.

[41] D. Stoller, S. Ewert, and S. Dixon, "Jointly Detecting and Separating Singing Voice: A Multi-Task Approach," in *Latent Variable Analysis and Signal Separation (LVA/ICA)*, Guildford, UK, 2018, pp. 329–339.

[42] R. Kumar, Y. Luo, and N. Mesgarani, "Music Source Activity Detection and Separation Using Deep Attractor Network," in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, Hyderabad, India, 2018, pp. 347–351.

[43] O. Slizovskaia, L. Kim, G. Haro, and E. Gomez, "End-to-end Sound Source Separation Conditioned on Instrument Labels," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, 2019, pp. 306–310.

[44] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic Music Transcription: An Overview," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 20–30, 2019.

[45] C.-W. Wu, "Addressing the data challenge in automatic drum transcription with labeled and unlabeled data," PhD Thesis, Georgia Institute of Technology, 2018.

[46] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. H. Engel, S. Oore, and D. Eck, "Onsets and Frames: Dual-Objective Piano Transcription," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018, pp. 50–57.

[47] J. Nam, J. Ngiam, H. Lee, and M. Slaney, "A Classification-based Polyphonic Piano Transcription Approach using Learned Feature Representations," *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 175–180, 2011.

[48] J. Thickstun, Z. Harchaoui, and S. Kakade, "Learning Features of Music From Scratch," in *Proceedings of the International Conference on Learning Representations, (ICLR)*, Toulon, France, 2017.

[49] C.-W. Wu, C. Dittmar, C. Southall, R. Vogl, G. Widmer, J. Hockman, M. Müller, and A. Lerch, "A Review of Automatic Drum Transcription," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 26, no. 9, pp. 1457–1483, 2018.

[50] T. Takahashi, S. Fukayama, and M. Goto, "Instrudive: A Music Visualization System Based on Automatically Recognized Instrumentation," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018, pp. 561–568.

[51] R. Caruana, "Multitask Learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.

[52] E. Manilow, P. Seetharaman, and B. Pardo, "Simultaneous separation and transcription of mixtures with multiple polyphonic and percussive instruments," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, formerly Barcelona, Spain, 2020, pp. 771–775.

[53] S. Böck, M. E. P. Davies, and P. Knees, "Multi-Task Learning of Tempo and Beat: Learning One to Improve the Other," in *Proceedings of the International Society*

for Music Information Retrieval Conference (ISMIR), Delft, The Netherlands, 2019, pp. 486–493.

[54] R. Vogl, M. Dorfer, G. Widmer, and P. Knees, "Drum Transcription via Joint Beat and Drum Modeling Using Convolutional Recurrent Neural Networks," in *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, China, 2017, pp. 150–157.

[55] Y.-N. Hung, Y.-A. Chen, and Y.-H. Yang, "Multitask Learning for Frame-level Instrument Recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, 2019, pp. 381–385.

[56] P. Herrera-Boyer, G. Peeters, and S. Dubnov, "Automatic classification of musical instrument sounds," *Journal of New Music Research*, vol. 32, no. 1, pp. 3–21, 2003.

[57] E. Benetos, M. Kotti, C. Kotropoulos, J. J. Burred, G. Eisenberg, M. Haller, and T. Sikora, "Comparison of Subspace Analysis-Based and Statistical Model-Based Algorithms for Musical Instrument Classification," in *Proceedings of the Workshop On Immersive Communication And Broadcast Systems*, Berlin, Germany, 2005.

[58] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The Million Song Dataset," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Miami, FL, USA, 2011, pp. 591–596.

[59] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, "Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset," in *Proceedings of the International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA, 2019.

[60] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *Proceedings of the International Conference on Learning Representations (ICLR)*, Banff, AB, Canada, 2014.

[61] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-Supervised Learning with Deep Generative Models," in *Advances in Neural Information Processing Systems (NeurIPS)*, Curran Associates, Inc., 2014, pp. 3581–3589.

[62] A. Tarvainen and H. Valpola, "Mean Teachers are Better Role Models: Weight-averaged Consistency Targets Improve Semi-Supervised Deep Learning Results," in *Advances in Neural Information Processing Systems (NeurIPS)*, Curran Associates, Inc., 2017, pp. 1195–1204.

[63] M. Sajjadi, M. Javanmardi, and T. Tasdizen, "Regularization With Stochastic Transformations and Perturbations for Deep Semi-Supervised Learning," in *Advances in*

Neural Information Processing Systems (NeurIPS), Curran Associates, Inc., 2016, pp. 1163–1171.

[64] K. D. Martin and Y. E. Kim, "Musical Instrument Identification: A Pattern-Recognition Approach," *The Journal of the Acoustical Society of America*, vol. 104, no. 3, pp. 1768–1768, 1998.

[65] N. H. Fletcher and T. D. Rossing, *The Physics of Musical Instruments*. Springer Science & Business Media, 2012.

[66] V. Lostanlen, J. Andén, and M. Lagrange, "Extended Playing Techniques: The next Milestone in Musical Instrument Recognition," in *Proceedings of the International Conference on Digital Libraries for Musicology (DLfM)*, Paris, France, 2018, pp. 1–10.

[67] G. Agostini, M. Longari, and E. Pollastri, "Musical Instrument Timbres Classification with Spectral Features," *EURASIP Journal on Advances in Signal Processing*, vol. 2003, no. 1, p. 943 279, 2003.

[68] L.-F. Yu, L. Su, and Y.-H. Yang, "Sparse Cepstral Codes and Power Scale for Instrument Identification," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, 2014, pp. 7460–7464.

[69] Y. Han, S. Lee, J. Nam, and K. Lee, "Sparse Feature Learning for Instrument Identification: Effects of Sampling and Pooling Methods," *The Journal of the Acoustical Society of America*, vol. 139, no. 5, pp. 2290–2298, 2016.

[70] J. Eggink and G. J. Brown, "Instrument Recognition in Accompanied Sonatas and Concertos," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, Montreal, QC, Canada, 2004, pp. 217–220.

[71] A. Livshin and X. Rodet, "Musical Instrument Identification in Continuous Recordings," in *Digital Audio Effects 2004*, Naples, Italy, 2004, pp. 1–6.

[72] F. Fuhrmann, M. Haro, and P. Herrera, "Scalability, Generality and Temporal Aspects in Automatic Recognition of Predominant Musical Instruments in Polyphonic Music.," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Kobe, Japan, 2009, pp. 321–326.

[73] J. J. Bosch, J. Janer, F. Fuhrmann, and P. Herrera, "A Comparison of Sound Segregation Techniques for Predominant Instrument Recognition in Musical Audio Signals," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Porto, Portugal, 2012, pp. 559–564.

[74] T. Kitahara, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, "Instrument Identification in Polyphonic Music: Feature Weighting to Minimize Influence of Sound Overlaps," *EURASIP Journal on Applied Signal Processing*, vol. 2007, no. 1, pp. 155–155, 2007.

[75] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC Music Database: Music Genre Database and Musical Instrument Sound Database," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Baltimore, MD, USA, 2003, pp. 229–230.

[76] A. Klapuri, "Multiple Fundamental Frequency Estimation by Summing Harmonic Amplitudes," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Victoria, Canada, 2006, pp. 216–221.

[77] T. Heittola, A. Klapuri, and T. Virtanen, "Musical Instrument Recognition in Polyphonic Audio using Source-filter Model for Sound Separation," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Kobe, Japan, 2009, pp. 327–332.

[78] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello, "MedleyDB: A Multitrack Dataset for Annotation-Intensive MIR Research," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, Taiwan, 2014, pp. 155–160.

[79] S. Gururani and A. Lerch, "Mixing Secrets: A Multi-track Dataset for Instrument Detection in Polyphonic Music," in *Late Breaking Demo (Extended Abstract), Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, China, 2017.

[80] *MUMS : McGill University Master Samples*, Montreal, QC, Canada, 1987.

[81] L. Fritts, *The University of Iowa Electronic Music Studios Musical Instrument Samples*, 1997.

[82] E. Humphrey, S. Durand, and B. McFee, "OpenMIC-2018: An Open Dataset for Multiple Instrument Recognition," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018, pp. 438–444.

[83] A. Kumar and B. Raj, "Audio Event Detection Using Weakly Labeled Data," in *Proceedings of the ACM International Conference on Multimedia (ACM MM)*, Amsterdam, The Netherlands, 2016, pp. 1038–1047.

[84] A. Bergamo and L. Torresani, "Exploiting Weakly-Labeled Web Images to Improve Object Classification: A Domain Adaptation Approach," in *Advances in Neural*

Information Processing Systems (NeurIPS), Curran Associates, Inc., 2010, pp. 181–189.

[85] D. Little and B. Pardo, "Learning Musical Instruments from Mixtures of Audio with Weak Labels," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Philadelphia, PA, USA, 2008, pp. 127–132.

[86] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "FMA: A Dataset for Music Analysis," in *Proceedings of the International Society for Music Information Retrieval Conference, (ISMIR)*, Suzhou, China, 2017, pp. 316–323.

[87] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. Weiss, and K. Wilson, "CNN Architectures for Large-Scale Audio Classification," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, USA, 2017, pp. 131–135.

[88] K. Choi, G. Fazekas, and M. Sandler, "Automatic Tagging using Deep Convolutional Neural Networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, New York, NY, USA, 2016, pp. 805–811.

[89] E. Cakır, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 25, no. 6, pp. 1291–1303, 2017.

[90] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[91] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Proceedings of the International Conference on Machine Learning (ICML)*, vol. 37, Lille, France: PMLR, 2015, pp. 448–456.

[92] D. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," in *Proceedings of the International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico, 2016.

[93] Z.-H. Zhou and M.-L. Zhang, "Neural Networks for Multi-Instance Learning," in *Proceedings of the International Conference on Intelligent Information Technology*, Beijing, China, 2002, pp. 455–459.

[94] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and Classification of Acoustic Scenes and Events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, 2015.

[95] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," in *Proceedings of the International Conference on Learning Representations, (ICLR)*, San Diego, CA, USA, 2015.

[96] Q. Kong, Y. Xu, W. Wang, and M. Plumbley, "Audio Set Classification with Attention Model: A Probabilistic Perspective," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, BC, Canada, 2018, pp. 316–320.

[97] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio Set: An Ontology and Human-Labeled Dataset for Audio Events," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, USA, 2017, pp. 776–780.

[98] Q. Kong, C. Yu, Y. Xu, T. Iqbal, W. Wang, and M. D. Plumbley, "Weakly labelled audioset tagging with attention neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 27, no. 11, pp. 1791–1802, 2019.

[99] B. McFee, J. Salamon, and J. P. Bello, "Adaptive Pooling Operators for Weakly Labeled Sound Event Detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 26, no. 11, pp. 2180–2193, 2018.

[100] Z.-H. Zhou and M.-L. Zhang, "Multi-Instance Multi-Label Learning with Application to Scene Classification," in *Advances in Neural Information Processing Systems (NeurIPS)*, Curran Associates, Inc., 2007, pp. 1609–1616.

[101] Z.-H. Zhou, M.-L. Zhang, S.-J. Huang, and Y.-F. Li, "Multi-Instance Multi-Label Learning," *Artificial Intelligence*, vol. 176, no. 1, pp. 2291 –2320, 2012.

[102] Z.-J. Zha, X.-S. Hua, T. Mei, J. Wang, G.-J. Qi, and Z. Wang, "Joint Multi-Label Multi-Instance Learning for Image Classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Anchorage, AK, USA, 2008, pp. 1–8.

[103] J. Foulds and E. Frank, "A Review of Multi-Instance Learning Assumptions," *The Knowledge Engineering Review*, vol. 25, no. 1, pp. 1–25, 2010.

[104] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola, "Deep Sets," in *Advances in Neural Information Processing Systems (NeurIPS)*, Curran Associates, Inc., 2017, pp. 3391–3401.

[105] T. Durand, N. Mehrasa, and G. Mori, "Learning a Deep ConvNet for Multi-Label Classification With Partial Labels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, pp. 647–657.

[106] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proceedings of the International Conference on Learning Representations, (ICLR)*, San Diego, CA, USA, 2015.

[107] M.-L. Zhang, Y.-K. Li, X.-Y. Liu, and X. Geng, "Binary Relevance for Multi-Label Learning: An Overview," *Frontiers of Computer Science*, vol. 12, no. 2, pp. 191–202, 2018.

[108] S. Mishra, B. L. Sturm, and S. Dixon, "Local Interpretable Model-Agnostic Explanations for Music Content Analysis," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, China, 2017, pp. 537–543.

[109] ——, "Understanding a Deep Machine Listening Model Through Feature Inversion," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018, pp. 755–762.

[110] R. Kelz and G. Widmer, "Towards Interpretable Polyphonic Transcription with Invertible Neural Networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, 2019, pp. 376–383.

[111] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. P. Vlahavas, "Multi-Label Classification of Music into Emotions," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Philadelphia, PA, USA, 2008, pp. 325–330.

[112] M.-L. Zhang and K. Zhang, "Multi-Label Learning by Exploiting Label Dependency," in *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD)*, Washington, DC, USA, 2010, pp. 999–1008.

[113] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, T. Mei, and H.-J. Zhang, "Correlative Multi-Label Video Annotation," in *Proceedings of the ACM International Conference on Multimedia (ACM MM)*, Augsburg, Germany, 2007, pp. 17–26.

[114] W. Bi and J. T. Kwok, "Multilabel Classification with Label Correlations and Missing Labels," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Quebec City, QC, Canada, 2014, pp. 1680–1686.

[115] H.-F. Yu, P. Jain, P. Kar, and I. S. Dhillon, "Large-scale Multi-label Learning with Missing Labels," in *Proceedings of the International Conference on Machine Learning (ICML)*, Beijing, China, 2014, pp. I–593–I–601.

[116] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel, "ReMixMatch: Semi-Supervised Learning with Distribution Matching and Augmentation Anchoring," in *Proceedings of the International Conference on Learning Representations (ICLR)*, Addis Adaba, Ethiopia, 2020.

[117] R. Fergus, Y. Weiss, and A. Torralba, "Semi-Supervised Learning in Gigantic Image Collections," in *Advances in Neural Information Processing Systems (NeurIPS)*, Curran Associates, Inc., 2009, pp. 522–530.

[118] M. Shi and B. Zhang, "Semi-Supervised Learning Improves Gene Expression-based Prediction of Cancer Recurrence," *Bioinformatics (Oxford, England)*, vol. 27, pp. 3017–23, 2011.

[119] Y. Liu and K. Kirchhoff, "Graph-Based Semisupervised Learning for Acoustic Modeling in Automatic Speech Recognition," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 24, no. 11, pp. 1946–1956, 2016.

[120] C. Rosenberg, M. Hebert, and H. Schneiderman, "Semi-supervised Self-Training of Object Detection Models," in *Proceedings of the IEEE Workshops on Applications of Computer Vision (WACV/MOTION)*, vol. 1, Breckenridge, CO, USA, 2005, pp. 29–36.

[121] A. Blum and T. Mitchell, "Combining Labeled and Unlabeled Data with Co-Training," in *Proceedings of the Annual Conference on Computational Learning Theory (COLT)*, Madison, Wisconsin, USA, 1998, pp. 92–100.

[122] K. Nigam and R. Ghani, "Analyzing the Effectiveness and Applicability of Co-Training," in *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, McLean, Virginia, USA, 2000, pp. 86–93, ISBN: 978-1-58113-320-2.

[123] A. Blum, J. Lafferty, M. R. Rwebangira, and R. Reddy, "Semi-supervised Learning Using Randomized Mincuts," in *Proceedings of the International Conference on Machine Learning (ICML)*, Banff, AB, Canada, 2004, pp. 13–.

[124] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised Learning Using Gaussian Fields and Harmonic Functions," in *Proceedings of the International Conference on Machine Learning (ICML)*, Washington, DC, USA, 2003, pp. 912–919.

[125] M. A. Ranzato and M. Szummer, "Semi-supervised Learning of Compact Document Representations with Deep Networks," in *Proceedings of the International Conference on Machine Learning (ICML)*, Helsinki, Finland, 2008, pp. 792–799.

[126] L. Maaløe, C. K. Sønderby, S. K. Sønderby, and O. Winther, "Auxiliary Deep Generative Models," in *Proceedings of the International Conference on Machine Learning (ICML)*, New York, NY, USA, 2016, pp. 1445–1454.

[127] A. Diment, T. Heittola, and T. Virtanen, "Semi-Supervised Learning for Musical Instrument Recognition," in *21st European Signal Processing Conference (EUSIPCO 2013)*, Marrakech, Morocco, 2013, pp. 1–5.

[128] S. Z. K. Khine, T. L. Nwe, and H. Li, "Singing Voice Detection in Pop Songs Using Co-Training Algorithm," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Las Vegas, NV, USA, 2008, pp. 1629–1632.

[129] G. Meseguer-Brocal, A. Cohen-Hadria, and G. Peeters, "DALI: A Large Dataset of Synchronized Audio, Lyrics and notes, Automatically Created using Teacher-student Machine Learning Paradigm.," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018, pp. 431–437.

[130] C.-W. Wu and A. Lerch, "From Labeled to Unlabeled Data – On the Data Challenge in Automatic Drum Transcription," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018, pp. 445–452.

[131] W. You and R. Dannenberg, "Polyphonic Music Note Onset Detection Using Semi-Supervised Learning.," in *Proceedings of the International Society for Music Information Retrieval Conference, (ISMIR)*, Vienna, Austria, 2007, pp. 279–282.

[132] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, *WaveNet: A Generative Model for Raw Audio*, 2016. arXiv: `1609.03499 [cs.SD]`.

[133] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, "GANSynth: Adversarial neural audio synthesis," in *Proceedings of the International Conference on Learning Representations (ICLR*, New Orleans, LA, USA, 2019.

[134] O. Luaces, J. Díez, J. Barranquero, J. J. del Coz, and A. Bahamonde, "Binary Relevance Efficacy for Multilabel Classification," *Progress in Artificial Intelligence*, vol. 1, no. 4, pp. 303–313, 2012.

[135] P. Bachman, O. Alsharif, and D. Precup, "Learning with Pseudo-Ensembles," in *Advances in Neural Information Processing Systems (NeurIPS)*, Curran Associates, Inc., 2014, pp. 3365–3373.

[136] S. Laine and T. Aila, "Temporal Ensembling for Semi-Supervised Learning," in *Proceedings of the International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.

[137] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "MixMatch: A Holistic Approach to Semi-Supervised Learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, Curran Associates, Inc., 2019, pp. 5049–5059.

[138] V. Verma, A. Lamb, J. Kannala, Y. Bengio, and D. Lopez-Paz, "Interpolation Consistency Training for Semi-supervised Learning," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, Macao, China, 2019, pp. 3635–3641.

[139] T. Miyato, S. Maeda, M. Koyama, and S. Ishii, "Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, pp. 1979–1993, 2019.

[140] B. Athiwaratkun, M. Finzi, P. Izmailov, and A. G. Wilson, "There Are Many Consistent Explanations of Unlabeled Data: Why You Should Average," in *International Conference on Learning Representations (ICLR)*, Vancouver, BC, Canada, 2018.

[141] M. Hyun, J. Jeong, and N. Kwak, *Class-Imbalanced Semi-Supervised Learning*, 2020. arXiv: 2002.06815 [cs.LG].

[142] M. Won, A. Ferraro, D. Bogdanov, and X. Serra, "Evaluation of CNN-based Automatic Music Tagging Models," in *Proceedings of the Sound and Music Computing (SMC)*, Torino, Italy, 2020, pp. 331–337.

[143] M. Won, S. Chun, O. Nieto, and X. Serrc, "Data-Driven Harmonic Filters for Audio Representation Learning," in *Proceedings of theIEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, 2020, pp. 536–540.

[144] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems (NeurIPS)*, Curran Associates, Inc., 2017, pp. 5998–6008.

[145] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, *FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence*, 2020. arXiv: 2001.07685 [cs.LG].

[146] R. Mignot and G. Peeters, "An Analysis of the Effect of Data Augmentation Methods: Experiments for a Musical Genre Classification Task," *Transactions of the International Society for Music Information Retrieval (TISMIR)*, vol. 2, no. 1, 2019.

[147] J. Schlüter and T. Grill, "Exploring Data Augmentation for Improved Singing Voice Detection with Neural Networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Málaga, Spain, 2015, pp. 121–126.